BS-SC: An Unsupervised Approach for Detecting Shilling Profiles in Collaborative Recommender Systems

Hongyun Cai and Fuzhi Zhang

Abstract—Collaborative recommender systems are vulnerable to shilling attacks. To address this issue, many methods including supervised and unsupervised have been proposed. However, supervised detection methods require training classifiers and they only apply to detect known types of attacks. The existing unsupervised detection methods need to know the prior knowledge of attacks, otherwise they suffer from low detection precision. In this paper, we present BS-SC, an unsupervised approach for detecting shilling profiles, which does not need to know the attack size or to label the candidate spammers. BS-SC starts from an in-depth analysis of user behaviors and uses two key mechanisms (i.e., behavior features extraction and behavior similarity matrix clustering) to distinguish shilling profiles from genuine ones. The behavior features reflect the behavior difference between genuine and shilling profiles, and the behavior similarity matrix clustering is to cluster shilling profiles based on their highly similar behaviors. Experimental results on the MovieLens and the sampled Amazon review datasets indicate that BS-SC outperforms the baseline unsupervised approaches, even when the prior knowledge is given for them.

Index Terms—Collaborative recommender systems, shilling attacks detection, behavior analysis, spectral clustering

1 Introduction

Collaborative recommender systems can model and push information actively for users based on their interest, which are widely used in many areas, e.g., e-commerce, cloud computing, social network, et al. [1]. With the coming of big data era, recommender systems will be paid more attention because the information overload problem will become more serious. However, different users have different purposes in interacting with recommender systems. Attackers can fabricate a large number of fake profiles (also called shilling profiles) in an attempt to promote or demote the recommend rank for the specific item. Such behavior has been referred to as shilling attacks or profile injection attacks [2], [3]. A number of studies have indicated that collaborative recommender systems are vulnerable to shilling attacks [4], which can be illustrated by an example.

Consider a movie recommender system that makes recommendation using the user-based collaborative algorithm. The system creates user profiles according to their ratings (in the scale of 1-5 and 5 indicates the most liked) on various movies. Table 1 shows seven genuine profiles (u1-7) and three shilling profiles (a1-3) injected by an attacker, as well as the Pearson correlation with u1. The purpose of the attacker is to promote the recommendation of movie6. Without the shilling profiles, u5 is the most similar user to u1 and the prediction for movie6 to u1 would be 2, indicating that the system will not recommend movie6 to u1. After the attack, however, a1 becomes the most similar user to u1 and movie6 will be recommended to u1 for a predicted rating of 5, indicating that the attacker can manipulate the recommendation by injecting shilling profiles.

In order to reduce the impact of shilling attacks on collaborative recommender systems, many methods have been proposed to detect shilling profiles over the past decade, which have greatly improved the level of shilling detection technology. Meanwhile, the attack models are evolving constantly in order to increase the difficulty of shilling detection [5], [6]. Moreover, new types of attacks will continue to appear [7], [8], [9]. In such cases, the existing shilling detection methods face the following challenges.

1) Supervised and semi-supervised detection methods require training classifiers on a training set, which suffer from low accuracy for misclassifying some genuine profiles as shilling ones when the attack types are unknown.
2) Unsupervised detection methods can detect shilling profiles without training classifiers, but most of them require the prior knowledge of attacks, such as the attack size and the candidate spammers, otherwise their detection performance is suboptimal. While the

- *H. Cai is with the School of Information Science and Engineering, Yanshan University, Qinhuangdao 066000, China, and also with the School of Cyber Security and Computer, Hebei University, Baoding 071000, China. E-mail: chy_hbu@126.com.*
- *F. Zhang is with the School of Information Science and Engineering, Yanshan University, Qinhuangdao 066000, China, and also with the Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao 066000, China. E-mail: xjzfz@ysu.edu.cn.*

Manuscript received 28 May 2018; revised 18 July 2019; accepted 2 Oct. 2019. Date of publication 8 Oct. 2019; date of current version 5 Mar. 2021.
(Corresponding author: Fuzhi Zhang.)
Recommended for acceptance by W. Zhang.
Digital Object Identifier no. 10.1109/TKDE.2019.2946247

TABLE 1
An Example of a Shilling Attack for Promoting Movie6

	movie1	movie2	movie3	movie4	movie5	movie6	correlation
u1	5		4	3	2	?	
u2	2		2	3			-0.87
u3	5	4	5			2	0.71
u4	3	5	4	2	2	1	0.67
u5	5	4	4	2	2	2	0.95
u6		3		1	2		-0.89
u7		4	2	2	3		-0.76
a1	5	4	4	3		5	1.00
a2	5		3	2		5	0.94
a3	5		3		3	5	0.78

prior knowledge required in these methods is difficult to acquire in practice.

To address the above problems, we present an unsupervised approach for detecting shilling profiles in collaborative recommender systems, which is called BS-SC. The presented approach focuses on extracting detection features from a new perspective of human behavior, which is different from the existing studies and can reveal the difference between genuine and shilling profiles in nature. Unlike the existing unsupervised detection approaches, our approach does not need to know the attack size, nor does it need to label the candidate spammers.

The main contributions of the paper are four-fold:

1) To extract the detection features that can reflect the behavior differences of genuine and attack users, we present the concept of user rating track which corresponds to the spatial trajectory of a user in the recommender system.
2) Inspired by the theory of differential encoding in digital coding system, we construct differential sequences corresponding to rating tracks and propose the feature extraction algorithm of preference stability degree.
3) We build multi-dimensional behavior vector for each user and calculate the behavior similarity between vectors based on euclidean distance and Gaussian kernel function, then we present a spectral clustering based shilling detection algorithm which does not need to know the attack size or to label the candidate spammers.
4) We carry out experiments on the MovieLens and sampled Amazon review datasets to compare the detection performance of BS-SC with the baseline unsupervised detection methods.

The rest of the paper is organized as follows. The background and related work are introduced in Section 2. In Section 3, we present our detection approach including the framework, behavior features extraction and the spectral clustering based shilling detection algorithm. In Section 4, experimental results are reported and analyzed. Conclusion and future work are given in the last section.

2 Background and Related Work

2.1 Background

The goal of a shilling attack in a CF recommender system is to promote or demote the recommendation of target items. To do so, a set of shilling profiles need to be injected into the system by the attacker. The strength of a shilling attack is usually relate to the number of shilling profiles and the number of rated items for each shilling profile, i.e., attack size and filler size. In general, a shilling profile consists of ratings on I_S, I_F, $I_\emptyset$, and I_t, which denote the set of selected items, the set of filler items, the set of unrated items, and the set of target items, respectively [4], [9].

The details of attack models used in this paper are described as follows:

Random attack: $I_S = \emptyset$ and I_F contains those randomly chosen filler items from $I - I_t$. The rating value for each filler item in I_F follows a normal distribution $N(\bar{r}, \bar{\sigma})$, where $\bar{r}$ and $\bar{\sigma}$ denote the mean and standard deviation over all items, respectively.

Average attack: Different with random attack, the rating value on each filler item follows $N(\bar{r_i}, \bar{\sigma_i})$, where $\bar{r_i}$ and $\bar{\sigma_i}$ are the mean and standard deviation of ratings for item p_i.

Bandwagon attack: I_S is chosen from popular items and their ratings are given the maximum rating value. I_F is composed of those randomly chosen filler items from $I - I_S - I_t$. The rating value given on each filler item follows $N(\bar{r}, \bar{\sigma})$.

Average-target shift attack: It is an obfuscating attack based on the average attack. The rating value on the target item is set to $r_{max} - 1$ for a push attack or $r_{min} + 1$ for a nuke attack.

Average-noise injecting attack: It is also an obfuscating attack based on the average attack. The rating value assigned to a filler item is added a Gaussian distribution random noise.

2.2 Related Work

The vulnerabilities of collaborative recommender systems to shilling attacks have led to a large number of studies focusing on detecting shilling attacks over the past decade. Under the hypothesis that all profiles are known types of samples, different supervised detection methods have been proposed. In [10], [11], some statistical metrics are presented by analysing the rating patterns of shilling profiles. These metrics are effective if shilling profiles are high-density while they are less effective for small-scale attacks. Yang et al. [12] extracted features based on the statistical properties of various attacks and applied a variant of AdaBoost as the classifiers, which improved the classification accuracy compared to a single classifier. Wu et al. [13] proposed a detector based on feature selection, which first used a feature selection algorithm to select effective features for detecting a specific type of attack, and then trained the classifier based on supervised learning. This method can improve the detection performance for known types of attacks. In addition, [14], [15] described a semi-supervised detector for hybrid shilling attacks, or HySAD for short, which worked with both labeled and unlabeled profiles, but only is effective for detecting a mixture of average and random attacks. Li et al. [16] proposed a shilling detection algorithm based on popularity degree features which started from the way users choose items to rate and trained the classifier using decision tree. [17], [18] examined the problem of shilling attacks detection from rating item distribution and presented two supervised detection models on the basis of novelty- and popularity-based item rating series. Zhou [19] used the theory of term frequency inverse document frequency to extract the features of AoP attack and

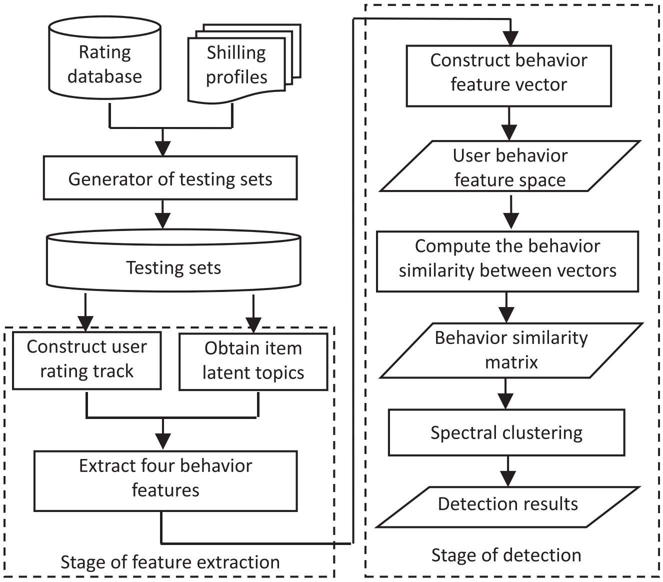

Rating database
Shilling profiles
Generator of testing sets
Testing sets
Construct user rating track
Obtain item latent topics
Extract four behavior features
Stage of feature extraction
Construct behavior feature vector
User behavior feature space
Compute the behavior similarity between vectors
Behavior similarity matrix
Spectral clustering
Detection results
Stage of detection

Fig. 1. Framework of BS-SC.

used a SVM-based classifier to detect shilling profiles generated by AoP model. In [20], Zhou et al. presented a two-stage model to detect shilling attacks by combining SVM-based method and target item analysis.

Unlike supervised or semi-supervised detection methods, unsupervised approaches do not require training classifier on labeled profiles. Zhang et al. [21] detected shilling profiles by analysing time series of ratings for each item. Mehta et al. [22] presented PCA-VarSelect which utilized principal component analysis to distinguish shilling profiles from genuine ones. PCA-VarSelect can detect standard attacks efficiently, but it requires to know the number of shilling profiles injected (i.e., the attack size), which is difficult to acquire accurately in practice. Lee et al. [23] thought that the location of effective shilling profiles should be the center of the distribution of them in order to influence most of genuine ones, so they proposed a hybrid two-phase detection approach based on Multidimensional Scaling (MDS). This approach first selected a subset of profiles that had high relationship with other profiles, and then discriminated shilling ones using a clustering-based method. The MDS-based approach is suitable for detecting average attack with high filler size, but it is less effective for detecting random attack with low filler size. Zou et al. [24] utilized the Belief Propagation algorithm to detect shilling attacks in collaborative filtering, which suffers from performance loss when the number of target items is small. Zhang et al. [25], [26] put forward graph-based detection algorithms for shilling attacks, which are ineffective for detecting obfuscated attacks. In [27], a framework based on fraudulent action propagation was introduced, which adopted a unified framework to detect shilling profiles regardless of the specific attack type. However, it needs to select some candidate spammers and the number of labeled spammers has a great influence on its detection performance. Zhou et al. [28] used statistical metrics of rating pattern and group characteristics to detect shilling profiles in recommender system. In addition, Zhou et al. [29] also utilized multi-dimension time series for detecting shilling attacks. Xia et al. [30] and G ü nnemann et al. [31] paid attention to detecting target items in a group of shilling profiles. Yang et al. [32] proposed a unsupervised detection method, which detected the shilling profiles by using the similarity of topological structure of attack users in the graph. Recently, Yang et al. [33] exploited a density-based clustering method to detect suspicious users, which clustered suspected users based on features extracted from item distribution and detected suspicious users using target item analysis.

TABLE 2
Notations and Their Descriptions

Notation	Description
$U=\{u_1, u_2, \ldots, u_k, \ldots, u_m\}$	user set in the recommender system, where u_k $(1 \leq k \leq m)$ is the kth user in U and $\|U\| = m$
$I=\{p_1, p_2, \ldots, p_j, \ldots, p_n\}$	item set in the recommender system, where p_j $(1 \leq j \leq n)$ is the jth item in I and $\|I\| = n$
$T = \left[t_{kj}\right]_{m \times n}$	rating time matrix, where t_{kj} denotes the time when u_k rated p_j
$R = \left[r_{kj}\right]_{m \times n}$	rating matrix, where r_{kj} is the rating by u_k on p_j
I_{u_k}	the set of items rated by u_k
$Track_k$	rating track of u_k
$Jac_{i,j}$	correlation degree between items p_i and p_j

3 The Proposed Approach

To improve the performance of shilling detection, we propose a novel behavior-based unsupervised approach for detecting shilling profiles. We first extract four detection features to characterize the behavior differences of genuine and shilling profiles, and then distinguish shilling profiles from genuine ones by presenting a spectral clustering based shilling detection algorithm.

As depicted in Fig. 1, the framework of BS-SC consists of two stages: feature extraction and detection. In the stage of feature extraction, we first construct rating track of each user and obtain latent topics of each item, then we extract four behavior features from the view of human behavior. In the stage of detection, behavior feature vector of every user is first constructed, which consists of the four extracted detection features, then the behavior similarity between vectors is calculated and the behavior similarity matrix is generated, and finally the detection results are obtained by performing spectral clustering on this similarity matrix. The details of BS-SC framework will be discussed in the following sections.

3.1 Notations

To facilitate the discussions, we give the descriptions of notations used in this paper in Table 2.

3.2 Feature Extraction

Human behavior is a complex process involved with intention, context, content, social, and et al. [34]. The indepth understanding of human behaviors is undoubtedly important in almost all human-related application scenarios, which can help to explain a lot of socio-economic phenomena. Although quantitative research on behavior dynamics is still in the stage of exploration, empirical researches [35], [36], [37], [38] on large-scale online data have uncovered

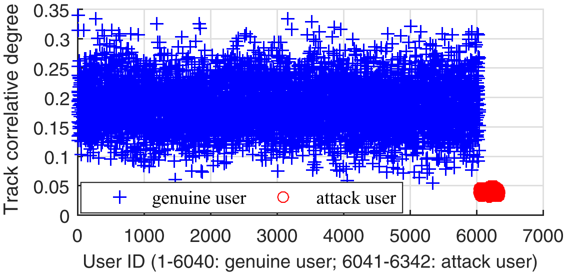


Fig. 2. Track correlative degree of genuine and attack users.

fat-tailed distributions on temporal and spatial activities of human behaviors, and pointed that the basic characteristics of human-interest dynamics include returning to previously visited interests preferentially, inertial effect, and exploration of new interests [37]. In recommender systems, the behaviors of genuine users are usually driven by their interests, thus evolution of interest behaviors has the characteristics of long-range correlation and preference stability. By contrast, the attack users usually rate filler items at random from those non-target items for lack of real consumption motivation, which leads to a significant difference in many aspects compared with genuine users. Accordingly, we measure these characteristics and extract four behavior features which can reflect obvious difference between genuine and shilling profiles.

3.2.1 Measurement of Long-Range Correlation on Interest

User interests are shown by the rated items in recommender systems, so coherence between interests is concordant with the association between the corresponding rated items. In this section, Jaccard similarity coefficient is used to measure the association between items, then we use track correlative degree to describe the long-range correlation on interest.

Definition 1 (User rating track). *For any user $u_k \in U$, user rating track of u_k is an order sequence of I_{u_k}, which is sorted according to the rating time of items rated by u_k in ascending order, and denoted as*

$$Track_k = \{p_{v_1}, p_{v_2}, \ldots, p_{v_s}\}, \tag{1}$$

where s is the length of the order sequence s.t. $s = |I_{u_k}|$, and p_{v_i} $(1 \leq i \leq s)$ is the ith item on $Track_k$ s.t. $t_{k,v_1} \leq t_{k,v_2} \leq \ldots \leq t_{k,v_s}$.

Definition 2 (Correlation coefficient between items). *For any two items $p_i \in I$ and $p_j \in I$, the correlation coefficient between p_i and p_j is calculated as follows*

$$Jac_{i,j} = \frac{\sum_{k=1}^{m} \mu_{k,i}\mu_{k,j}}{\sqrt{\sum_{k=1}^{m} \mu_{k,i}}^2 + \sqrt{\sum_{k=1}^{m} \mu_{k,j}}^2 - \sum_{k=1}^{m} (\mu_{k,i}\mu_{k,j})}, \tag{2}$$

where $\mu_{k,i}$ denotes whether u_k has rated item p_i, i.e., $\mu_{k,i} = \begin{cases} 1, & r_{ki} \neq \phi \\ 0, & r_{ki} = \phi \end{cases}$ and s.t. $1 \leq i, j \leq n$.

Jaccard similarity coefficient is used to calculate the value of $Jac_{i,j}$. It can be seen from Eq. (2) that $Jac_{i,j}$ is relevant to the number of users who have co-rated items p_i and p_j. And specifically, the more the number of co-rated users, the larger the value of $Jac_{i,j}$.

Definition 3 (Track correlative degree). *For any user $u_k \in U$, the track correlative degree of u_k refers to the degree of long-rang correlations on $Track_k$, which is defined as follows*

$$Track\_corr_k = \frac{1}{|Track_k|} \sum_{l=1}^{|Track_k|} Jac_{v_l, v_{l+1}}, \tag{3}$$

where $|Track_k|$ is the cardinality of $Track_k$, $Jac_{v_l,v_{l+1}}$ is the correlation coefficient between the lth and (l+1)th item on $Track_k$.

Fig. 2 illustrates the difference in track correlative degree for genuine and attack users. The genuine users are selected from the MovieLens 1M dataset. The attack ones are generated by average attack with 5 percent attack size and 5 percent filler size. As shown in Fig. 2, values of track correlation degree for genuine users are larger than those of attack ones, indicating that there has obvious difference between genuine and attack users in track correlation degree. Therefore, track correlative degree is used as one of the four detection features in our BS-SC model.

3.2.2 Diversity Measurement of User's Interest

Diversity of user interest in recommender systems means the users varied tastes, indicated by the rated items. There are different sources for defining diversity, such as taxonomies of items [39], genres of items [40], and latent topics of items [40]. Considering that the number of items for each taxonomy or genre varies greatly, so latent topics of items will be used as the source of diversity in this paper, which can be obtained by latent factor model, e.g., Probabilistic Latent Semantic Analysis (PLSA) [41] and Latent Dirichlet Allocation (LDA) [42]. In BS-SC, we use the Gibbs LDA model to extract latent topics of each item, which employs the Gibbs sampling technique and has better representation ability than PLSA.

In the Gibbs LDA, Θ and Φ are the Dirichlet distributions, which represent the document (item) distribution over topics and the topic distribution over words (users), respectively. The generation process is as follows.

1) For each item p_i, generate a Dirichlet distribution Θ_i with parameter α.
2) For each topic x, generate a Dirichlet distribution Φ_x with parameter β.
3) For each rating behavior on item p_i given by user u, generate the multinomial distribution z_{tu} with parameter Θ_i, and generate the multinomial distribution of user u with parameter $\Phi_{z_{tu}}$, respectively.

The document distribution over the topics and the topic distribution over words can be estimated using the Gibbs sampling method. Hyper-parameters α and β generally have a smoothing effect on multinomial parameters. Lower values of α and β, more decisive topic distributions. In our work,α and β are set to 0.1 and 0.1, respectively. The number of topics can be decided according to the metric of perplexity [42]. By experiment, we set the number of topics to 20, 60, and 60 on the MovieLens 100K, MovieLens 1M and sampled Amazon review datasets, respectively. For more information about the Gibbs LDA, please refer to [42].

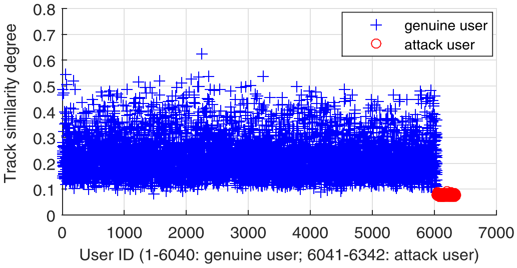


Fig. 3. Track similarity degree of genuine and attack users.

By using Gibbs LDA, each item $p_i \in I$ corresponds to a vector of latent topic distribution, denoted as $TopicV_i = (dp_{i,1}, dp_{i,2}, \ldots, dp_{i,q})$, where q means the total number of topics extracted from the rating matrix and $dp_{i,x}$ $(x = 1, 2, \ldots, q)$ means the probability of item p_i belonging to the xth topic.

Based on the latent topics of items, we present two new metrics (i.e., track similarity degree and interest uncertainty degree) to capture the diversity of user interest by using the correlations between vectors of latent topics.

Definition 4 (Track similarity degree). *For any user $u_k \in U$, the track similarity degree of user u_k measures the average interest similarity between items on $Track_k$, which is defined as follows*

$$Track\_sim_k = \frac{\sum_{p_{v_x} \in Track_k} \sum_{p_{v_y} \in Track_k, p_{v_x} \neq p_{v_y}} f_o(p_{v_x}, p_{v_y})}{|Track_k| \times (|Track_k| - 1)}, \quad (4)$$

where $f_o(p_{v_x}, p_{v_y})$ denotes the similarity between vectors of latent topic distributions corresponding to items p_{v_x} and p_{v_y}. Hereby, similarity may be calculated by various methods, e.g., Pearson correlation coefficient, euclidean distance, Cosine similarity, et al. In this paper, we use Cosine similarity which is widely used in many areas.

For track similarity degree of different users, lower value denotes higher diversity. Fig. 3 illustrates the track similarity degree of genuine and attack users. All genuine users are selected from the MovieLens 1M dataset. The attack ones are generated by the same method as that used in Section 3.2.1. The number of latent topics is 60. As shown in Fig. 3, the track similarity degree of attack users is lower than that of most genuine ones and concentrates on a small range,indicating that attack users' track similarity degree differ from that of genuine users. In addition, different users may have different range of interests. For example, some users' interests are more specific to few topics, while others may have broad interests across a wide range of topics. Here, we focus on the uncertainty of the item latent topics given a user profile.

As each item rated by user u_k corresponds to a vector of latent topic distribution, we can know the extent to which the rated item is associated with every latent topic. Therefore, for all items on $Track_k$, we use standard deviation to capture the uncertainty of user interest on each latent topic. Furthermore, the average of standard deviations on all latent topics is used to measure the users interest diversity or uncertainty.

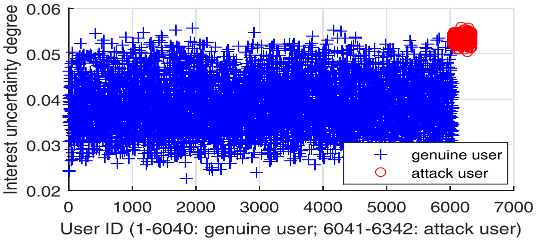


Fig. 4. Interest uncertainty degree of genuine and attack users.

Definition 5 (Interest uncertainty degree). *For any user $u_k \in U$, the interest uncertainty degree of user u_k, denoted as $Track\_unc_k$, refers to the uncertainty of the user interest indicated by the rated items of user u_k and calculated by*

$$Track\_unc_k = \frac{1}{q} \sum_{x=1}^{q} \sqrt{\frac{1}{|Track_k|} \sum_{v=1}^{|Track_k|} (dp_{v,x} - \mu_x)^2}, \quad (5)$$

where q means the total number of latent topics, $|Track_k|$ denotes the number of items rated by user u_k, $dp_{v,x}$ means the probability that the vth item on $Track_k$ belongs to the xth topic, and μ_x is the mean of probabilities corresponding to the xth topic. That is to say, the interest uncertainty degree of user u_k is calculated by the average of standard deviation of all latent topic distribution vectors corresponding to the items on $Track_k$.

For interest uncertainty degree of different users, higher value denotes more uncertainty or diversity of user interest. Fig. 4 shows the interest uncertainty degree for 6,040 genuine users and 302 attack ones. Here, the attack users are also generated by average attack with 5 percent attack size and 5 percent filler size. It can be seen from Fig. 4, the interest uncertainty degree of attack users is higher than that of most genuine users, indicating that the attack users' interests are more diverse.

It is noted that a user, whose interest is highly exclusive, might have a high interest similarity degree and a low interest uncertainty degree, whereas a user with a broad interest may be reversed. However, as shown in Figs. 3 and 4, both the track similarity degree and interest uncertainty degree of attack users are concentrated in a smaller scope, which reveals the high similarity between attack users in the aspect of interest diversity. Therefore, track similarity degree and interest uncertainty degree are used as detection features in BS-SC.

3.2.3 Stability Measurement of Users Interest

Memory is a key property of human behavior, so interest may be stability in a short period and people often have more tendencies on the recent hobby because of memory and inertial effect, though skipping from one hobby to another is also common in the sequence of human behavior. Inspired by the theory of differential encoding in digital coding system, we put forward a stability measurement mechanism of user interest preference, which will be discussed in the following.

In latent factor model, topics do not usually define disjoint or isolated categories. That is to say, every item may belong to one or more latent topics. To facilitate the calculation, we

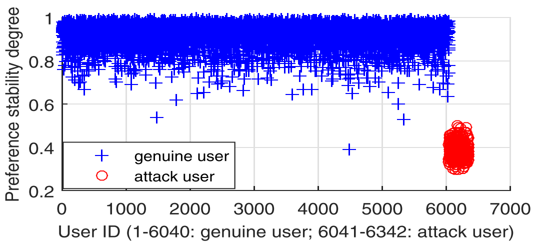


Fig. 5. Preference stability degree of genuine and attack users.

generate a boolean matrix $I\_G = [ig_{i,j}]_{n\times q}$ on the basis of latent factor model, where the ith row corresponds to item p_i, the jth column refers to the jth latent topic, and $ig_{i,j}$ indicates whether or not item p_i belongs to the jth latent topic. Here, $ig_{i,j} = 1$ means that item p_i belongs to the jth latent topic; otherwise, $ig_{i,j} = 0$.

Definition 6 (Consistency of interest preferences). *For any user $u_k \in U$, the consistency of interest preference for the* l*th and (l+1)th rating behaviors denotes the consistency of item topics on items p_{v_l} and $p_{v_{l+1}}$, which is defined by*

$$pre\_con_k^{l,l+1} = \begin{cases} 1, & \text{if } (\exists x)((ig_{v_l,x} == 1) \wedge (ig_{v_{l+1},x} == 1)) \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where $x \in \{1, 2, \ldots, q\}$ and $1 \leq l \leq |Track_k| - 1$.

As discussed above, consistency of interest preference can judge whether interest preferences have been changed on two adjacent rating items. Next, we can further analyse the change on the whole rating track.

Definition 7 (Consistency sequence of interest preference). *For any user $u_k \in U$ and $Track_k$, the consistency sequence of interest preference is defined as a binary sequence with the length of $|Track_k|$, and denoted by $jump\_seq_k$. Let $jump\_seq_k = a_1a_2 \ldots a_{|Track_k|}$ and a_1 be 1, the other element a_r ($1 \leq r \leq |Track_k|$) is defined as*

$$a_r = \begin{cases} a_{r-1}, & \text{if } pre\_con_k^{r-1,r} = 1 \\ a_{r-1} \oplus 1, & \text{otherwise} \end{cases}. \tag{7}$$

By the definition of $jump\_seq_k$, two adjacent elements are equal in $jump\_seq_k$ if interest preferences are the same one for the corresponding two adjacent rating behaviors on $Track_k$; otherwise not. That is to say, the more two adjacent elements are equal in $jump\_seq_k$, the more stable the interest preference is. Inspired by the theory of differential encoding in digital coding system, we can further construct a differential sequence for $jump\_seq_k$ and evaluate the stability of differential sequence.

Definition 8 (Differential sequence of interest preference). *For any user $u_k \in U$, the differential sequence of u_k is a binary sequence and denoted as $Diff\_seq_k = b_1b_2 \ldots b_{l'}$, the* i*th element is defined as*

$$b_i = \begin{cases} 1, & \text{if } a_i = a_{i+1} \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

where a_i and a_{i+1} are the i*th and (i+1)th elements of $jump\_seq_k$ s.t. $1 \leq i \leq l'$ and $l' = |Track_k| - 1$.*

Definition 9 (Preference stability degree). *For any user $u_k \in U$, preference stability degree of u_k denotes the stability degree of interest preference on $Track_k$, which is defined as*

$$pre\_sta_k = \frac{\|Diff\_seq_k\|_0}{|Diff\_seq_k|}, \tag{9}$$

where $\|Diff\_seq_k\|_0$ is the L0-norm of $Diff\_seq_k$, $|Diff\_seq_k|$ is the cardinality of $Diff\_seq_k$.

Fig. 5 illustrates the distribution of preference stability degree of 6,342 users which include 6,040 genuine users and 302 attack ones. The genuine users are from the MovieLens 1M dataset. The attack ones are generated by the same method as that used in Section 3.2.1. It can be seen from Fig. 5, preference stability degrees of attack users are concentrated in a smaller range. Moreover, the preference stability degree of attack user is lower than that of most genuine ones. Hence we also use preference stability degree as behavior feature in our BS-SC model.

Based on the above definitions and analysis, the feature extraction algorithm for preference stability degree is described as follows.

Algorithm 1. Feature Extraction of Preference Stability Degree

```
Require: rating time matrix $T$, item-topic boolean matrix $I\_G$
Ensure: preference stability degree of any user $u_k \in U$
 1: $P\_S = \phi$
 2: for $\forall u_k \in U$ do
 3:     Obtain $I_{u_k}$
 4:     $Track_k \leftarrow$ sort $I_{u_k}$ by rating time in ascending order
 5:     $jump\_seq_k = \phi, Diff\_seq_k = \phi, jump\_seq_k[1] \leftarrow 1$
 6:     for e=2 to $|Track_k|$ do
 7:         compute $pre\_con_k^{e-1,e}$ according to Eq. (6)
 8:         compute $jump\_seq_k[e]$ according to Eq. (7)
 9:         compute $Diff\_seq_k[e-1]$ according to Eq. (8)
10:     end for
11:     $P\_S[k] \leftarrow (\|Diff\_seq_k\|_0/|Diff\_seq_k|)$
12: end for
13: return $P\_S$
```

The major computational burden of Algorithm 1 is to generate rating track and construct differential sequence for each user. Given item set I and user $u_k \in U$, the time complexity of generating $Track_k$ (lines 3 and 4) is $O(|I|) + O(|I_{u_k}| \times \log_2|I_{u_k}|)$, where O(n) denotes the complexity of calculating the set of items rated by user u_k, $O(|I_{u_k}| \times \log_2|I_{u_k}|)$ is the complexity of sorting I_{u_k}. The time complexity of constructing differential sequence (lines 5 to 10) is $3O(1) + 5(|Track_k| - 1)O(1)$. Since $|I_{u_k}| \ll |I|$, the total time complexity of computing pre_sta_k is $O(|I|) + O(|I_{u_k}| \times \log_2|I_{u_k}|) + 3O(1) + 5(|Track_k| - 1)O(1) + O(1) \approx O(|I|)$.

3.3 Shilling Profiles Detection

An individual shilling profile has little impact on the target item while a group of shilling profiles can bring remarkable effect. To produce better attack effect, a group of shilling profiles may be injected into the rating database simultaneously [1], which manifests high similarity on their behavior features for the same intention. So, we can cluster shilling profiles together due to their high behavior similarity.

3.3.1 Constructing Behavior Similarity Matrix

To compute the behavior similarity between users, we first give the definition of behavior feature vector, and then we use Gaussian kernel function to calculate the behavior similarity on the basis of the behavior distance between behavior feature vectors.

Definition 10 (Behavior feature vector). *For any user $u_k \in U$, the behavior feature vector of u_k is a four-tuple*

$$BFV_k = (Track\_corr_k, Track\_sim_k, Track\_unc_k, pre\_sta_k), \tag{10}$$

where $Track\_corr_k$ is the track correlation degree, $Track\_sim_k$ is the interest similarity degree, $Track\_unc_k$ is the interest uncertainty degree, and pre_sta_k is the preference stability degree.

Let $BFS = \{BFV_k | k = 1, 2, \ldots, |U|\}$ be the set of behavior feature vectors, the less the distance between two vectors in *BFS*, the smaller the behavior difference between them. According to the pattern recognition theory, nonlinear mapping can change linearly inseparable pattern in the low dimensional space into a separable one in a high dimensional space. To cluster shilling profiles and avoid the curse of dimensionality in the high dimensional space, we compute behavior similarity between users based on Gaussian kernel function (also known as *RBF* kernel function).

To neutralize the effect of the range of values for different behavior features, we normalize the values of behavior features by using Min-Max normalization. More specifically, the normalized feature $f_{k,i} = \frac{f'_{k,i} - \min(f_i)}{\max(f_i) - \min(f_i)}$, where ${f_{k,i}}'$ denotes the *i*th feature of user u_k, $\max(f_i)$ and $\min(f_i)$ are the maximum and minimum values of the *i*th behavior feature, respectively.

Definition 11 (Behavior distance). *For any two users $u_k, u_{k'} \in U$, the behavior distance between BFV_k and $BFV_{k'}$ is denoted by $bd_{k,k'}$, which is computed according to the weighted euclidean distance and defined as*

$$bd_{k,k'} = \sqrt{\sum_{i=1}^{s} (w_i \cdot (f_{k,i} - f_{k',i}))^2}, \tag{11}$$

where s *is the number of components of a behavior feature vector, s.t.* s=4 *in BS-SC. $f_{k,i}$ and $f_{k',i}$ denote the* i*th normalized component of BFV_k and $BFV_{k'}$ respectively, and w_i is the weight coefficient of the* i*th component that is determined based on the importance of component contribution to the behavior difference.*

Through the analysis in Section 3.2, we observe that there are obvious differences in the extracted features between genuine profiles and shilling ones. Therefore, we set the weight coefficient of four features to the same value.

Definition 12 (Behavior similarity). *For any $u_k, u_{k'} \in U$, the behavior similarity between them is defined*

$$bs_{k,k'} = \exp\left(-\frac{bd_{k,k'}}{2\sigma^2}\right)/mbs, \tag{12}$$

where $bd_{k,k'}$ is the behavior distance between u_k and $u_{k'}$, σ is the scale parameter of Gaussian kernel function, mbs *denotes the maximum of all behavior similarity. On the basis of behavior similarity, we can generate the behavior similarity matrix which is a symmetric matrix and denoted by $B\_S = \left[bs_{k,k'}\right]_{m \times m}$.*

3.3.2 Clustering on Behavior Similarity Matrix

The behavior similarity matrix can be viewed as a weighted undirected graph $G =< V, E >$, where the vertices of graph G correspond to the users in the recommender system, and an edge is formed between any pair of vertices. The weight on each edge is the behavior similarity between the corresponding two users. As mentioned before, shilling profiles have high behavior similarity with the other ones, but they have large behavior distance with genuine profiles. So we can distinguish shilling profiles from genuine ones by the optimal partition algorithm in graph theory, which can maximize the intra-group similarities and minimize the inter-group similarities.

As optimum solution for graph partition is NP-complete, optimal partition can be solved by the Fiedler vector using a continuous relaxation spectrum. For Laplacian matrix of graph, the second smallest eigenvalue is called the Fiedler value, and the corresponding eigenvector is called Fiedler vector which gives information for partitioning the graph. This kind of methods, based on strict mathematical analysis and spectral graph theory, is called as spectral clustering algorithm.

Standard spectral clustering algorithm (e.g., NCut and RCut) [43] can partition a graph into two roughly equal-sized sub-graphs based on Eq. (13).

$$S_*(C_S, \overline{C_S}) = \arg\min_S \left(\sum_{\substack{u \in C_S, v \in \overline{C_S} \\ (u,v) \in E}} w(u,v)\right)\left(\frac{size(V)}{size(C_S)} + \frac{size(V)}{size(\overline{C_S})}\right), \tag{13}$$

where S stands for a cut that partitions V into C_S and $\overline{C_S}$, $S_*(C_S, \overline{C_S})$ is the optimal partitions, u and v denote vertices in C_S and $\overline{C_S}$ respectively, $w(u,v)$ is the weight between u and v, $size(C) = \sum_{u \in C, v \in V} w(u,v)$ for NCut and $size(C) = |C|$ for RCut.

In recommender systems, considering attack cost and difficulty of attack detection, the number of shilling profiles is usually less than that of genuine ones. Therefore, the problem of shilling attack detection is corresponding to unbalanced clustering with upper bounds of size constraints. Here, we use PCut as spectral clustering algorithm to deal with unbalanced data, which was proposed in [43] and gave the optimal partitions described by

$$S_*(C^*, \overline{C^*}) = \arg\min_S \sum_{\substack{u \in C_S, v \in \overline{C_S} \\ (u,v) \in E}} w(u,v)|\delta|V| \; < \; \min\{|C_S|, |\overline{C_S}|\} \leq \frac{1}{2}|V|, \tag{14}$$

where $S_*(C^*, \overline{C^*})$ stands for the optimal partitions under the constraint condition, δ is a non-negative constant corresponding to the lower bound of size constraint. It is noted that BS-SC focuses only on the partition with upper bound constrained. Therefore, parameter δ may be ignored here.

The main steps of clustering algorithm based on PCut are described below.

(1) Rank Computation

The rank $R(v_i)$ of any $v_i \in V$ is calculated by

$$\mathrm{R}(v_i) = \frac{1}{|V|} \sum_{v_j \in V} \Gamma\{\eta(v_i) \geq \eta(v_j)\}, \tag{15}$$

where Γ denotes the indicator function described as Eq. (16), $\eta(v_i)$ is a density function reflecting the relative density at v_i and described as Eq. (17).

$$\Gamma = \begin{cases} 1, & \text{if } \eta(v_i) \geq \eta(v_j) \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$\eta(v_i) = \frac{1}{|N(v_i)|} \sum_{x \in N(v_i)} bs_{x,v_i}. \tag{17}$$

In Eq. (17), $N(v_i)$ is the set of *top_k* nearest neighbors of v_i, which consists of those vertices having higher weights on the edges linked to v_i, so $\eta(v_i)$ is the average behavior similarity of *top_k* nearest neighbors. Here, it is noted that the value of *top_k* should not be larger than the number of vertices in the smaller sub-graph. Otherwise, the clustering accuracy will be reduced.

(2) Graph Construction

For any vertex $v_i \in V$ and edges connected to v_i, weights of the edges related to $k_\lambda(v_i)$ nearest neighbors remain unchanged while those of other edges are set to zero. $k_\lambda(v_i)$ is calculated by

$$k_\lambda(v_i) = a(\lambda + 2(1-\lambda)\mathrm{R}(v_i)), \tag{18}$$

where λ(Lamda) is a scalar parameter to handle unbalanced clustering and its value is in the range of 0 and 1,*a* is an integer for controlling the number of nearest neighbors.

(3) Graph Separation

Let G' be the reconstruction of graph G, e_1 and e_2 are the corresponding eigenvectors for the first and second smallest eigenvalue of Laplacian Matrix of G', respectively. Two sub-graphs can be separated based on k-means clustering in the feature space $Q = [e_1, e_2]$, the profiles in the sub-graph with a smaller size are viewed as shilling ones.

3.3.3 Spectral Clustering Based Shilling Detection Algorithm

Based on Sections 3.3.1 and 3.3.2, the main steps of shilling detection in BS-SC are as follows:

1) Construct behavior similarity matrix;
2) Reconstruct behavior graph based on rank computation;
3) Cluster shilling profiles by spectral clustering algorithm;

According to the above steps, the spectral clustering based shilling detection algorithm is described as follows.

Algorithm 2. Spectral Clustering based Shilling Detection

```
Require: $BFS = \{BFV_k | k = 1, 2, \ldots, |U|\}$, values of $\sigma$, $\lambda$, $a$ and $top\_k$
Ensure: the set of shilling profiles $C_f$
1: for $\forall u_k, u_{k'} \in U$ do
2:     compute $bd_{k,k'}$ according to Eq. (11)
3:     $bs'_{k,k'} \leftarrow \exp(-\frac{bd_{k,k'}}{2\sigma^2})$
4: end for
5: $mds \leftarrow \max_{1 \leq k,k' \leq |U|} bs_{k,k'}$
6: for $\forall k, k' \in [1, |U|]$ do
7:     $bs_{k,k'} \leftarrow \frac{bs'_{k,k'}}{mds}$
8: end for
9: for $i = 1$ to $|U|$ do
10:    compute $R(v_i)$ and $k_\lambda(v_i)$ by Eqs. (15) and (18)
11: end for
12: for $\forall k, k' \in [1, |U|]$ do
13:    $x_k \leftarrow$ the minimum among $k_\lambda(v_\lambda)$ nearest neighbors
14:    if $bs_{k,k'} < x_k$ then
15:        $bs_{k,k'} \leftarrow 0$
16:    end if
17: end for
18: calculate graph degree matrix D
19: calculate normalized Laplacian matrix $L \leftarrow D^{-1/2}(IM - B\_S) D^{-1/2}$
20: $Q \leftarrow$ the second smallest eigenvector of L
21: cluster users in U into $C_1$ and $C_2$ by K-means algorithm
22: $C_f \leftarrow$ the smaller group in $C_1$ and $C_2$
23: return $C_f$
```

Algorithm 2 mainly includes three parts. The first part, from lines 1 to 8, is to construct behavior similarity matrix, the time complexity is $O(|U|^2)$. The second part, from lines 9 to 17, is to reconstruct the behavior similarity graph aiming at the problem of unbalanced clustering, and the time complexity is $O(|U|^2)$. The third part, from lines 18 to 23, is to distinguish shilling profiles from genuine ones by using spectral clustering algorithm. Specifically, line 18 is to calculate the degree matrix D corresponding to the behavior similarity graph, which is a diagonal matrix and the *i*th diagonal element is the sum of the elements on *i*th row of B_S. Line 19 is to generate normalized Laplacian matrix *L*, where $D^{-1/2}$ denotes the inverse matrix of $D^{1/2}$ and $D^{1/2}$ is the square roots of D, *IM* is a unit matrix with the same order comparing to B_S. Line 20 is to compute the second smallest eigenvector of L which includes the information for graph partitioning. Line 21 is to divide G' into two sub-graphs by using *K*-means algorithm. The profiles in the smaller cluster are viewed as shilling ones (line 22). The time complexity of the third part is $O(|U|^2) + O(|U|^3) + O(|U|^3) + O(|U| + O(|U| \approx O(|U|^3))$. Therefore, the major computational burden of Algorithm 2 is spectral clustering, and the total time complexity of this algorithm is $O(|U|^2) + O(|U|^2) + O(|U|^3) \approx O(|U|^3))$.

4 Experiment And Evaluation

4.1 Experimental Data and Settings

To evaluate the effectiveness of BS-SC, we carry out experiments on synthetic and real datasets. Detection results are averaged over 10 trials.

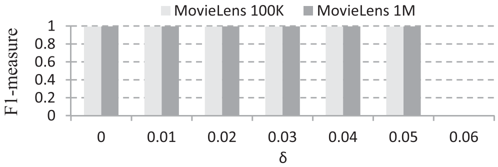


Fig. 6. The influence of parameter δ on the F1-measure of BS-SC.

1) Synthetic datasets. We generate two synthetic datasets based on the MovieLens 100K and 1M datasets,[1] respectively. The MovieLens 100K dataset contains 100,000 rating records on 1,682 items from 943 users. The MovieLens 1M dataset contains 1,000,209 rating records on 3,952 items from 6,040 users. All ratings are integer values in the range of 1 and 5, where 1 and 5 denote most disliked and liked, respectively. Every user has rated at least 20 movies in both datasets. Rating time is UNIX seconds since 1/1/1970 UTC. Similar to the previous studies, all profiles on the MovieLens datasets are viewed as genuine ones. Shilling profiles are generated and injected under various attack models described in Section 2.1 (i.e., random attack, average attack, bandwagon attack, average-target shift attack and average-noise injecting attack), respectively. The attack size is set to 5 and 15 percent, respectively. The filler size is set to 5, 10, 15, and 20 percent, respectively. For a push attack, the target item is randomly selected from unpopular items.
2) Real dataset. We use a sampled Amazon review dataset [44] as the experimental real dataset, which consists of 53,777 rating records given by 5,055 labeled users to 17,610 items. All ratings are integer between 1 (most disliked) and 5 (most liked). The original Amazon dataset was obtained by crawling from Amazon.cn. It contains 1,205,125 ratings on 136,785 items from 645,072 users. The sampled Amazon dataset is created for evaluation on the original dataset, which is extracted through frequent itemset mining, online discussions and careful observations. Among 5055 users on the sampled dataset, 3,118 users are labeled as genuine and 1,937 users are attackers. Considering that there is no enough information to infer user's exact taste with a small number of rated items, we filter out those users whose number of rated items is less than 5.

4.2 Evaluation Metrics

We use precision, recall, and F1-measure as evaluation metrics. These metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

$$Recall = \frac{TP}{P} \tag{20}$$

$$F1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{21}$$

where TP is the number of shilling profiles correctly detected, FP is the number of genuine ones misclassified as shilling profiles, P is the total number of shilling profiles in the recommender system.

4.3 Experimental Results and Analysis

In this section, we compare the precision and recall of BS-SC with the following methods.

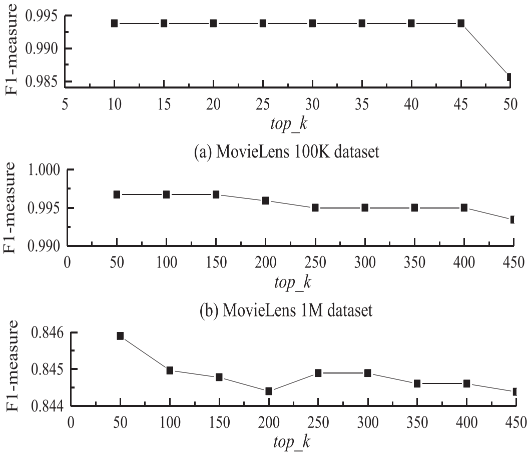
1) PCA-VarSelect: The PCA-VarSelect method proposed in [22]. This method is a classical unsupervised attack detection algorithm, which can achieve favorable detection performance provided that the number of shilling profiles has been known. In contrast experiments, we assume that the prior knowledge is known in advance.
2) CBS: The Catch the Black Sheep method proposed in [27]. This method is a representative approach of detecting shilling profiles on real dataset. It is regardless of the specific attack models, but its detection performance is affected by the labeled candidate spammers and the estimated number of spammers. In the experiments, we assume that the number of spammers is known in advance. In addition, 10 percent shilling profiles are selected as candidate spammers under each attack size on the MovieLens datasets. For the experiments on the sampled Amazon review dataset, we randomly choose 10 labeled shilling profiles as the candidate spammers.
3) EUB-DAR: An unsupervised detection method proposed in [32]. This method is a two-stage detection framework without prior knowledge of attack size, which detects shilling profiles using their similarity of topological structure in the graph. In the experiments, parameters t and ε are set to 30 and 20, respectively.

4.3.1 *Parameter Selection*

In BS-SC, parameters include δ, σ, top_k, a, and λ. The value of scale parameter σ is related to the performance of Gaussian kernel function. In this paper, we set σ to 0.5 according to [45]. To illustrate the influence of parameters δ, top_k, a, and λ, we inject the shilling profiles generated by random attack model with 5 percent attack size and 5 percent filler size into the MovieLens 100K and 1M datasets, respectively. All the shilling profiles are push attacks, but it is noted that there is no difference between push and nuke attacks in BS-SC. Figs. 6, 7, and 8 show the influence of parameters δ, top_k, a, and λ on the F1-measure of BS-SC, respectively.

It can be seen from Fig. 6, BS-SC can obtain excellent performance when the value of δ is less than 0.05. The values of F1-measure of BS-SC are over 0.98 and there are no great difference between them when $\delta \leq 0.05$. However, if we set δ to a larger value, the cluster containing shilling profiles will be discarded because it is not satisfied with the lower bound of size constraint. This is why the F1-measure of BS-SC will decrease to zero when δ is set to 0.06. In the paper, we do not require the prior knowledge of attack size and only focus on the partition with upper bound constrained, i.e., the cluster with smaller size by PCut is identified as the set of shilling profiles no matter what size it is. Therefore, parameter δ is set to zero on the experimental datasets.

[1] https://grouplens.org/datasets/movielens/

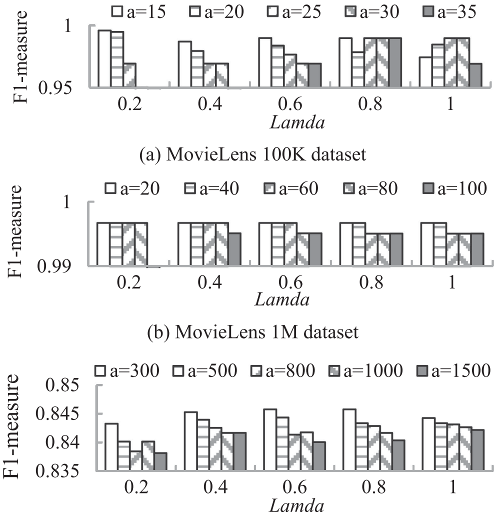


(a) MovieLens 100K dataset

(b) MovieLens 1M dataset

(c) Sampled Amazon dataset

Fig. 7. The influence of parameter top_k on the F1-measure of BS-SC.



(a) MovieLens 100K dataset

(b) MovieLens 1M dataset

(c) Sampled Amazon dataset

Fig. 8. The influence of parameters *a* and λ(Lamda) on the F1-measure of BS-SC. Note that the F1-measure values below the minimum ordinate value are not be displayed.

It is observed from Fig. 7a, except for $top\_k = 50$, the F1-measure of BS-SC has no significant difference for various values of top_k on the MovieLens 100K dataset. As shown in Figs. 7b and 7c, the F1-measure of BS-SC declines slightly with the increasing of top_k on the Movielens 1M and sampled Amazon datasets. These results indicate that genuine users may be misclassified as attack ones in the case of a larger value of parameter top_k. In the experiments, we set parameter top_k to 20, 100, and 50 on the MovieLens 100K dataset, MovieLens 1M dataset, and Amazon dataset, respectively.

Fig. 8 shows the influence of parameters *a* and λ on the F1-measure of BS-SC, where top_k is set to 20 on the MovieLens 100K dataset, 100 on the MovieLens 1M dataset and 50 on the Amazon dataset. As shown in Fig. 8a, the F1-measure of BS-SC is more than 0.95 for various values of a when parameter λ is greater than 0.4 on the MovieLens 100K dataset. When $\lambda = 0.2$ and $\lambda = 0.4$, the detection performance of BS-SC will decrease significantly if parameter a increases to some extent (i.e., $a = 30$ for $\lambda = 0.2$ or $a = 35$ for $\lambda = 0.4$). It can be seen from Fig. 8b, under 5 percent attack size on the MovieLens 1M dataset, the F1-measure of BS-SC is almost 1 when parameter λ is greater than 0.2. When $\lambda = 0.2$ and $a = 100$, BS-SC has low F1-measure. As shown in Fig. 8c, the F1-measure values of BS-SC on the Amazon dataset are between 0.8382 and 0.8458, indicting that parameters *a* and λ have very slight impact on the F1-measure of BS-SC. The reason is that the behavior similarity matrix is sparse on the Amazon dataset and it does not change much at the step of graph construction. Based on the description in Section 3.3.2, the value of $a(\lambda + 2(1 - \lambda)R(v_i))$ should not be larger than the total number of shilling profiles, otherwise some genuine profiles may be misclassified as shilling ones, which can decrease the detection performance of BS-SC. Therefore, λ is set to greater than 0.4 and *a* should not exceed the number of shilling profiles. In the experiments, we set parameters λ and *a* to 0.8 and 15 on the MovieLens 100K dataset, 0.6 and 60 on the MovieLens 1M dataset, 0.8 and 300 on the Amazon dataset, respectively.

TABLE 3
F1-Measure of Seven Methods on Three Datasets

Methods	MovieLens 100K	MovieLens 1M	Sampled Amazon review dataset
TCD-SC	0.9519	0.9961	0.4924
TSD-SC	0.9895	0.9923	0.8435
IUD-SC	0.9475	0.2385	0.0226
PSD-SC	0.9215	0.9947	0.1441
TCD+TSD	0.9899	0.9962	0.8440
TCD+TSD+PSD	0.9910	0.9964	0.8441
BS-SC	0.9919	0.9964	0.8447

4.3.2 Analysis of Contribution from Four Features

To illustrate the effectiveness of the contribution from four features, we conduct experiments to compare the methods with different number of features to the approach with all four features (i.e., BS-SC) on two synthetic datasets and the sampled Amazon review dataset, respectively. For ease of description, we denote four methods with only individual feature (i.e., track correlation degree, track similarity degree, interest uncertainty degree, and preference stability degree) as TCD-SC, TSD-SC, IUD-SC, and PSD-SC, respectively. Two methods with two and three of the most effective features on three datasets are denoted as TCD+TSD and TCD+TSD+PSD, respectively.

On the MovieLens 100K dataset, we inject attack profiles under five attack models with 5 percent attack size and various filler sizes (i.e., 5, 10, 15, and 20 percent), respectively. On the MovieLens 1M dataset, we inject attack profiles under five attack models with 5 percent filler size and 5 percent attack size. Each experiment was repeated 5 times. The results are averaged on each dataset. The results of contrast experiments for seven methods are listed in Table 3.

As shown in Table 3, the F1-measure values of TCD-SC, TSD-SC, IUD-SC, and PSD-SC are over 0.92 on the MovieLens 100K dataset. This indicates the effectiveness of each feature on the MovieLens 100K dataset. It can be seen from Table 3, the F1-measure of TCD-SC, TSD-SC, and PSD-SC is over 0.99 on the MovieLens 1M dataset. This means almost all genuine profiles and attack ones can be distinguished accurately by each individual feature of track correlation

TABLE 4
Comparison between BS-SC and Other Methods for Detecting the Five Attacks at Various Attack Sizes and Various Filler Sizes on the MovieLens 100K Dataset

Attack model	Attack size	Filler size	PCA-VarSelect		CBS		EUB-DAR		BS-SC	
			Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Random attack	5%	5%	0.9413	0.9851	0.5455	0.5714	0.5116	0.9362	0.9877	1
		10%	0.9489	0.9894	0.7273	0.7619	0.4891	0.9574	0.9958	1
		15%	0.9510	0.9957	0.8409	0.8810	0.5181	0.9149	0.9958	1
		20%	0.9400	0.9830	0.9318	0.9763	0.3667	0.9362	1	1
	15%	5%	0.8704	1	0.8108	0.9400	0.5523	0.9469	0.9900	1
		10%	0.8698	0.9993	0.8581	1	0.7041	0.9787	0.9916	1
		15%	0.8704	1	0.8581	1	0.7394	0.9858	0.9972	1
		20%	0.8704	1	0.8582	1	0.6683	0.9787	1	1
Average attack	5%	5%	0.9103	0.9550	0.5102	0.5238	0.4231	0.9362	0.9833	1
		10%	0.9026	0.9381	0.8182	0.8571	0.3359	0.9362	1	1
		15%	0.8898	0.9277	0.9091	0.9524	0.3358	0.9574	1	1
		20%	0.8723	0.9085	0.9509	1	0.3538	0.9787	1	1
	15%	5%	0.8667	0.9957	0.8403	0.9800	0.5344	0.9362	0.9874	1
		10%	0.8685	0.9979	0.8598	1	0.6479	0.9787	0.9972	1
		15%	0.8660	0.9950	0.8602	1	0.6222	0.9574	1	1
		20%	0.8648	0.9936	0.8597	1	0.6402	0.9716	1	1
Bandwagon attack	5%	5%	0.8163	0.8511	0.4322	0.4511	0.4835	0.9361	0.9386	1
		10%	0.8980	0.9362	0.7958	0.8309	0.4831	0.9148	0.9773	1
		15%	0.8776	0.9149	0.9100	0.9021	0.5000	0.9574	0.9979	1
		20%	0.9388	0.9787	0.9100	1	0.4835	0.9362	1	1
	15%	5%	0.8333	0.9574	0.8533	0.9800	0.5865	0.9858	0.9718	1
		10%	0.8642	0.9929	0.8613	1	0.6186	0.9858	0.9860	1
		15%	0.8704	1	0.8636	1	0.6600	0.9716	0.9923	1
		20%	0.8704	1	0.8600	1	0.7128	0.9858	0.9965	1
Average-target shift attack	5%	5%	0.9203	0.9617	0.6123	0.6399	0.2659	0.9572	0.9833	1
		10%	0.8942	0.9320	0.7539	0.7852	0.4792	0.9575	1	1
		15%	0.9012	0.9400	0.8987	0.9500	0.4687	0.9780	1	1
		20%	0.8921	0.9260	0.9338	1	0.4399	0.9787	1	1
	15%	5%	0.8679	0.9970	0.8582	0.9805	0.6283	0.9858	0.9867	1
		10%	0.8679	0.9972	0.8603	1	0.6816	0.9716	0.9972	1
		15%	0.8675	0.9965	0.8622	1	0.6603	0.9787	1	1
		20%	0.8648	0.9963	0.8605	1	0.6692	0.9858	1	1
Average-noise injecting attack	5%	5%	0.9245	0.9638	0.5901	0.6211	0.3733	0.5512	0.9917	1
		10%	0.9060	0.9447	0.8225	0.8597	0.3958	0.7877	1	1
		15%	0.8959	0.9340	0.8599	0.9110	0.3807	0.7006	1	1
		20%	0.8878	0.9255	0.9300	0.9833	0.3303	0.7226	1	1
	15%	5%	0.8685	0.9979	0.8409	0.9760	0.3988	0.7092	0.9860	1
		10%	0.8690	0.9986	0.8620	1	0.4811	0.6300	0.9958	1
		15%	0.8685	0.9979	0.8633	1	0.4947	0.6672	0.9986	1
		20%	0.8673	0.9965	0.8633	1	0.5193	0.6669	1	1

degree, track similarity degree, and preference stability degree. However, the F1-measure of IUD-SC is only 0.2385 on the MovieLens 1M dataset, indicating that many genuine profiles are misclassified as attack ones only by the interest uncertainty degree. This is because many genuine users also present more uncertainty or diversity of user interest with a large number of latent topics (the number of latent topics is set to 60 on the MovieLens 1M dataset). It can also be seen from Table 3, on the sampled Amazon dataset, the F1-measure of TCD-SC, TSD-SC, IUD-SC and PSD-SC is 0.4924, 0.8435, 0.0226 and 0.1441, respectively, indicating that track correlation degree and track similarity degree are effective but interest uncertainty degree and preference stability degree are less effective. As the ratio of attack users to the total users is about 38 percent on this dataset and a number of genuine users are similar with attack users in their interest uncertainty degree under a large number of latent topics, these genuine and attack users may be divided into the sub-graph with a larger size. This is why the F1-measure of IUD-SC is very low on the Amazon dataset. The Amazon dataset is too sparse (the sparsity level is 99.94 percent) and many users only rated a few items, these users may be misclassified by PSD-SC. This is the reason that PSD-SC is less effective on the sampled Amazon dataset. In addition, as shown in Table 3, the F1-measure values of TCD+TSD, TCD+TSD+PSD and BS-SC are better than that of four methods with each individual feature on three datasets. Furthermore, it should be noted that the F1-measure of BS-SC (i.e., the approach with all four features) is always the best among these methods on three datasets.

TABLE 5
Comparison between BS-SC and Other Methods for Detecting the Five Attacks at 5 percent Filler Size Across Various Attack Sizes on the MovieLens 1M Dataset

Attack model	Attack size	PCA-VarSelect		CBS		EUB-DAR		BS-SC	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Random attack	5%	0.9401	0.9868	0.9473	1	0.4876	0.9801	0.9934	1
	15%	0.8703	1	0.8578	1	0.7431	0.9868	0.9967	1
Average attack	5%	0.9110	0.9563	0.9270	0.9298	0.5008	0.9735	0.9934	1
	15%	0.8697	0.9993	0.8587	1	0.7496	0.9746	0.9978	1
Bandwagon attack	5%	0.8612	0.9040	0.9521	1	0.4636	0.9274	0.9902	1
	15%	0.8592	0.9872	0.8582	1	0.7362	0.9426	0.9967	1
Average-target shift attack	5%	0.9117	0.9491	0.8815	0.9305	0.4702	0.9669	0.9934	1
	15%	0.8684	0.9990	0.8610	1	0.7388	0.9614	0.9978	1
Average-noise injecting attack	5%	0.9129	0.9511	0.8779	0.9260	0.4948	0.9801	0.9934	1
	15%	0.8699	0.9981	0.8581	1	0.7454	0.9792	0.9978	1

These results show that four features can make an effective contribution to the detection of shilling profiles.

4.3.3 Comparison of Detection Results on the MovieLens 100K Dataset

Table 4 shows the detection results of four methods with five attacks (i.e., random attack, average attack, bandwagon attack, average-target shift attack and average-noise injecting attack) at various attack sizes and filler sizes on the MovieLens 100K dataset.

As listed in Table 4, on the MovieLens 100K dataset, the detection precision of PCA-VarSelect under five attacks is between 0.8163 and 0.9510. The recall of PCA-VarSelect under five attacks is over 0.8511. This means that PCA-VarSelect can perform well in detecting five attacks with the prior knowledge of the number of shilling profiles. The reason is that PCA-VarSelect detects the shilling profiles based on the principal components of rating matrix. The detection precision and recall of CBS can be improved when the attack size and filler size are increased, but the values of precision and recall are only about 50 percent under 5 percent attack size and 5 percent filler size. CBS detects the shilling profiles based on the idea of label propagation, so the detection performance of CBS is regardless of the specific attacks. However, some genuine profiles will be identified as shilling ones when the number of candidate spammers is small. The recall values of EUB-DAR are over 0.9 in detecting random attack, average attack, bandwagon attack and average-target shift attack. The precision values of EUB-DAR are not high under five attacks, indicating that many genuine profiles are identified as shilling ones by EUB-DAR. Compared with three baselines, the detection performance of BS-SC is the best in detecting five attacks at various attack sizes and filler sizes. Its precision values are over 0.93 and all values of recall are 1, indicating that BS-SC can effectively detect various attacks. Moreover, the precision of BS-SC increases with the increasing of filler size no matter what attack models are used. This is because the extracted features are more effective for distinguishing attack users from genuine ones if attack users have rated more items. Therefore, we can conclude that BS-SC outperforms PCA-VarSelect, CBS, and EUB-DAR in terms of precision and recall metrics in detecting the five attacks at various attack sizes and filler sizes on the MovieLens 100K dataset.

4.3.4 Comparison of Detection Results on the MovieLens 1M dataset

To enhance the credibility of experimental results, contrast experiments are repeated on the MovieLens 1M dataset. The results of contrast experiments for the four methods (i.e., PCA-VarSelect, CBS, EUB-DAR and BS-SC) are shown in Table 5. Given that the average number of items rated by each user is 166 on the MovieLens 1M dataset, which is slightly less than 5 percent filler size, so we compare experimental results only under 5 percent filler size.

As shown in Table 5, the precision and recall of BS-SC is higher than that of PCA-VarSelect, CBS, and EUB-DAR in detecting the five attacks under 5 percent filler size across various attack sizes on the MovieLens 1M dataset. The values of precision for BS-SC approach to 1, while those of PCA-VarSelect and CBS are about 0.9 with the prior knowledge of attack size. It can be seen in Table 5, the recall values of BS-SC are 1. These results indicate that the detection performance of BS-SC is better than that of PCA-VarSelect, CBS, and EUB-DAR. The reason is that BS-SC utilizes obvious behavior differences between genuine and shilling profiles, as well as the highly similar behaviors among shilling profiles.

It can also be seen in Tables 4 and 5, under the same conditions (i.e., the attack, attack size, and filler size are the same), the precision and recall values of CBS and BS-SC on the MovieLens 1M dataset are better than those of them on the MovieLens 100K dataset. For CBS, in the case of the same attack size, the number of candidate spammers on the MovieLens 1M dataset is greater than that on the MovieLens 100K dataset, so its detection performance is improved as the number of candidate spammers increases. For BS-SC, in the case of the same filler size, the number of rated items for shilling profiles on the MovieLens 1M dataset is greater than that of those on the MovieLens 100K dataset, so the characteristics of shilling profiles are more apparent.

4.3.5 Comparison of Detection Results on the Sampled Amazon Review Dataset

Fig. 9 shows the relationship between the F1-measure of BS-SC and the number of latent topics on the sampled Amazon review dataset. It can be seen from Fig. 9, BS-SC is effective for detecting shilling profiles when the number of topics is no

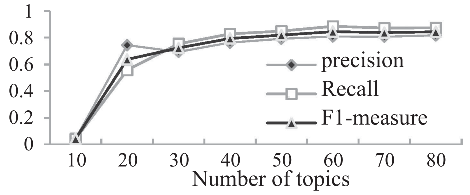


Fig. 9. Relationship between the F1-measure of BS-SC and the number of latent topics.

less than 20. The precision of BS-SC is o.7433, but the recall is only 0.5574 when the number of topics is set to 20. This indicates that a number of shilling profiles may be misclassified as genuine ones. The reason is that users interest preferences cannot be represented clearly with less latent topics on this dataset. With the increasing of the number of latent topics, the difference of track similarity degree between genuine users and attack ones becomes more obvious. Therefore, the detection performance of BS-SC will further be improved when the number of latent topics is larger. Moreover, the F1-measure of BS-SC shows no remarkable change when the number of topics is larger than 40. These results mean that BS-SC can reach better detection performance if the number of latent topics is set in a reasonable range.

We conduct experiments on the sampled Aamzon review dataset to compare BS-SC with three baselines. Fig. 10 shows the detection results of four methods.

It can be seen from Fig. 10, the detection performance of PCA-VarSelect and EUB-DAR is not good, and the F1-measure values of them are under 45 percent. PCA-VarSelect can discover some clusters of highly correlated shilling profiles. However, on the sampled Amazon dataset, attack users form many different groups and those in a loose group are not closely related to each other. Moreover, genuine users may be detected as attackers if some of their rated items are the target items of attack users. This is why the precision and recall of PCA-VarSelect is low on this dataset. EUB-DAR can detect attack users that have dense connections between each other. However, the sampled Amazon dataset is very sparse and many attack users only rate a few items. Moreover, there are close connections between some genuine users. Therefore, many genuine profiles and shilling profiles on this dataset cannot be classified precisely by EUB-DAR. The F1-measure of CBS is 52.33 percent, which is better than that of PCA-VarSelect and EUB-DAR, indicating that CBS is effective in detecting attacks on the Amazon dataset with the prior knowledge of attack size and some labeled candidate attack users. The sampled Amazon dataset holds sufficient colluders and different attack users may rate different target items. Therefore, some attack users may have high spam probability and can be detected if they have common target items with the labeled candidate attack users, while other attack users can not be detected by CBS. As shown in Fig. 10, the precision, recall and F1-measure of BS-SC can reach over 80 percent on this dataset, indicating that BS-SC can not only cluster most shilling profiles together but also distinguish shilling profiles from genuine ones. On the Amazon dataset, attack users have high behavior similarity no mather what their target items are. This is the reason that BS-SC is effective for detecting attacks on this real dataset. Therefore, we can conclude that BS-SC has excellent detection performance in detecting shilling profiles on the sampled Amazon dataset and it outperforms PCA-VarSelect, EUB-DAR and CBS.

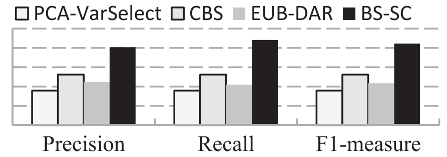


Fig. 10. Comparison of four methods on the sampled Amazon dataset.

5 Conclusions and Future Work

Detection of shilling profiles can help to improve the credibility of recommender systems. In this paper, we study the behavior characteristics of users and present a spectral clustering based unsupervised approach to detect shilling profiles. We start by the memory and inertial effect of human behaviors and extract four detection features reflecting behavior differences between genuine and shilling profiles, i.e., track correlation degree, track similarity degree, interest uncertainty degree, and preference stability degree. We further construct the behavior feature space composed by all behavior feature vectors, and calculate the behavior similarly between vectors based on the euclidean distance and Gaussian kernel function. We detect shilling profiles by using spectral clustering algorithm in the behavior feature space, which is based on highly similar behaviors among shilling profiles. Experimental results on the MovieLens and the sampled Amazon review datasets have shown the excellent performance of BS-SC in detecting shilling profiles.

This work is an attempt to detect shilling profiles from the view of human behavior analysis, and it needs further studies on human behavior analysis in recommender systems. Besides the numerical ratings and item topics that we used for extracting behavior features, we can incorporate more information available in recommender systems to model the behavior for genuine and shilling profiles. In our future research, we will focus on behavior evolution and quantitative analysis of evolutional rule in recommender systems, which can provide theoretical basis of human dynamics in detecting shilling profiles.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61772452, 61379116), the Natural Science Foundation of Hebei Province, China (Nos. F2015203046, F2014201165), and the Key Program of Research on Science and Technology of Higher Education Institutions of Hebei Province, China (No. ZD2016043).

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Berlin, Germany: Springer, 2015.

[2] S. K. Lam and J. Riedl, “Shilling recommender systems for fun and profit,” in *Proc. 13th Int. Conf. World Wide Web*, 2004, pp. 393–402.

[3] M. P. O’Mahony, N. J. Hurley, and G. C. M. Silvestre, “Recommender systems: Attack types and strategies,” in *Proc. 20th Nat. Conf. Artif. Intell.*, 2005, pp. 334–339.

[4] C. Li and Z. Luo, “Detection of shilling attacks in collaborative filtering recommender systems,” in *Proc. Int. Conf. Soft Comput. Pattern Recognit.*, 2011, pp. 190–193.

[5] C. Williams and B. Mobasher, “Profile injection attack detection for securing collaborative recommender systems,” DePaul University, Chicago, IL, Rep. no. 06-014, 2006.

[6] N. J. Hurley, Z. P. Cheng, and M. Zhang, "Statistical attack detection," in *Proc. 3rd Int. Conf. Recommender Syst.*, 2009, pp. 149–156.

[7] D. C. Wilson and C. E. Seminario, "When power users attack: Assessing impacts in collaborative recommender systems," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 427–430.

[8] C. E. Seminario and D. C. Wilson, "Nuke'Em till they go: Investigating power user attacks to disparage items in collaborative recommenders," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 293–296.

[9] C. E. Seminario and D. C. Wilson, "Attacking item-based recommender systems with power items," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 57–64.

[10] P. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proc. 7th Int. Workshop Web Inf. Data Manage.*, 2005, pp. 67–74.

[11] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Classification features for attack detection in collaborative recommendation systems," in *Proc. 12th Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 542–547.

[12] Z. Yang, L. Xu, Z. Cai, and Z. Xu, "Re-scale AdaBoost for attack detection in collaborative filtering recommender systems," *J. Knowl.-Based Syst.*, vol. 100, pp. 74–88, 2015.

[13] Z. Wu, Y. Zhuang, Y. Wang, and J. Cao, "Shilling attack detection based on feature selection for recommendation system," *J. Acta Electronica Sinica*, vol. 40, no. 8, pp. 1687–1693, 2012.

[14] J. Cao, Z. Wu, B. Mao, and Y. Zhang, "Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system," *J. World Wide Web*, vol. 16, no. 5/6, pp. 729–748, 2013.

[15] Z. Wu, J. Wu, and J. Cao, "HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 985–993.

[16] W. T. Li, M. Gao, H. Li, Q. Y. Xiong, J. H. Wen, and B. Ling, "An shilling attack detection algorithm based on popularity degree features," *J. Acta Automatica Sinica*, vol. 41, no. 9, pp. 1563–1575, 2015.

[17] F. Zhang and Q. Zhou, "HHT-SVM: An online method for detecting profile injection attacks in collaborative recommender systems," *J. Knowl.-Based Syst.*, vol. 65, no. 4, pp. 96–105, 2014.

[18] F. Zhang and H. Chen, "An ensemble method for detecting shilling attacks based on ordered item sequences," *J. Security Commun. Netw.*, vol. 9, no. 7, pp. 680–696, 2015.

[19] Q. Q. Zhou, "Supervised approach for detecting average over popular items attack in collaborative recommender systems," *J. IET Inf. Security*, vol. 10, no. 3, pp. 134–141, 2016.

[20] W. Zhou, J. Wen, Q. Xiong, M. Gao, and J. Zeng, "SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems," *Neurocomputing*, vol. 210, no. C, pp. 197–205, 2016.

[21] S. Zhang, A. Chakrabarti, J. Ford, and F. Makedon, "Attack detection in time series for recommender systems," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 809–814.

[22] B. Mehta and W. Nejdlk, "Unsupervised strategies for shilling detection and robust collaborative filtering," *J. User Model. User-Adapted Interact.*, vol. 19, no. 1/2, pp. 65–97, 2009.

[23] J. Lee and D. Zhu, "Shilling attack detection-A new approach for a trustworthy recommender system," *J. Informs J. Comput.*, vol. 24, no. 1, pp. 117–131, 2012.

[24] J. Zou and F. Fekri, "A belief propagation approach for detecting shilling attacks in collaborative filtering," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 1837–1840.

[25] Z. Zhang and S. R. Kulkarni, "Graph-based detection of shilling attacks in recommender systems," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2013, pp. 1–6.

[26] Z. Zhang and S. R. Kulkarni, "Detection of shilling attacks in recommender systems via spectral clustering," in *Proc. 17th Int. Conf. Inf. Fusion*, 2014, pp. 1–8.

[27] Y. Zhang, Y. Tan, M. Zhang, Y. Liu, T.-S. Chua, and S. Ma, "Catch the black sheep: Unified framework for shilling attack detection based on fraudulent action propagation," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2408–2414.

[28] W. Zhou, J. Wen, M. Gao, H. Ren, and P. Li, "Abnormal profiles detection based on time series and target item analysis for recommender systems," *J. Math. Problems Eng.*, vol. 2015, pp. 1–9, 2015.

[29] W. Zhou, J. Wen, Q. Xiong, J. Zeng, L. Liu, H. Cai, and T. Chen, "Ab-normal group user detection in recommender systems using multi-dimension time series," in *Proc. 12th Int. Conf. Collaborate Comput.: Netw. Appl. Work-Sharing*, 2016, pp. 373–383.

[30] H. Xia, B. Fang, M. Gao, H. Ma, Y. Tang, and J. Wen, "A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique," *J. Inf. Sci.*, vol. 306, no. C, pp. 150–165, 2015.

[31] N. Günnemann, S. Günnemann, and C. Faloutsos, "Robust multivariate autoregression for anomaly detection in dynamic product ratings," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 361–372.

[32] Z. Yang, Z. Cai, and X. Guan, "Estimating user behavior toward detecting anomalous ratings in rating systems," *Knowl.-Based Syst.*, vol. 111, pp. 144–158, 2016.

[33] Z. Yang, Z. Cai, and X. Guan, "Spotting anomalous ratings for rating systems by analyzing target users and items," *Neurocomputing*, vol. 240, pp. 25–46, 2017.

[34] P. Cui, H. Liu, C. Aggarwal, and F. Wang, "Uncovering and predicting human behaviors," *IEEE J. Intell. Syst.*, vol. 31, no. 2, pp. 77–88, Mar./Apr. 2016.

[35] M. Jiang, P. Cui, F. Wang, X. Xu, W. Zhu, and S. Yang, "FEMA: Flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1186–1195.

[36] Z. Zhao, S. Cai, J. Huang, Y. Fu, and T. Zhou, "Scaling behavior of online human activity," *J. Europhys. Lett.*, vol. 100, no. 4, 2012, Art. no. 48004.

[37] T. Zhou, X. Han, X. Yan, Z. Yang, Z. Zhao, and B. Wang, "Statistical mechanics on temporal and spatial activities of human," *J. University Electron. Sci. Technol. China*, vol. 42, no. 4, pp. 481–540, 2013.

[38] Z. D. Zhao, "Research on analysis, modeling and dynamics of spatial-temporal characteristics of human behaviors," PhD dissertation, School of Computer Science and Engineering, Univ. Electron. Sci. Technol. China, Chengdo, China, 2014.

[39] C. Ziegler, S. M. Mcnee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proc. 14th Int. Conf. World Wide Web*, 2005, pp. 22–32.

[40] S. Vargas, P. Castells, and D. Vallet, "Explicit relevance models in intent-oriented information retrieval diversification," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 75–84.

[41] Y. Kabutoya, T. Iwata, H. Toda, and H. Kitagawa, "A probabilistic model for diversifying recommendation lists," *Web Technol. Appl.*, vol. 7808 of LNCS, pp. 348–359, 2013.

[42] G. Heinrich, "Parameter estimation for text analysis," Technical report, Fraunhofer IGD, Darmstadt, Germany, May 2005.

[43] J. Qian and V. Saligrama, "Spectral clustering with imbalanced data," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 3057–3061.

[44] C. Xu, J. Zhang, C. Long, and C. Long, "Uncovering collusive spammers in Chinese review websites," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 979–988.

[45] J. Wang, T. Jebara, and S. F. Chang, "Graph transduction via alternating minimization," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 282–286.

Hongyun Cai received the MA degree in computer application technology from Hebei University, China, in 2005. She is working toward the PhD degree in the School of Information Science and Engineering, Yanshan University, China. From 2005 to date, she has worked in Hebei University, China, where she is currently an associate professor. Her research interests include information security and recommender systems.

Fuzhi Zhang received the PhD degree from the Beijing Institute of Technology, China, in 2003. From 1986 to date, he has worked at Yanshan University, China, where he is currently a professor and PhD supervisor. His main research interests include intelligent network information processing, information security, service-oriented computing, etc.