Semi-SAD: Applying Semi-supervised Learning to Shilling Attack Detection

Zhiang Wu[1], Jie Cao[1*], Bo Mao[2], Youquan Wang[1]
1. Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, P.R. China
2. Geoinformatics, Royal Institute of Technology, Stockholm, Sweden
zawuster@gmail.com, caojie690929@163.com, {maoboo, youq.wang}@gmail.com

ABSTRACT

Collaborative filtering (CF) based recommender systems are vulnerable to shilling attacks. In some leading e-commerce sites, there exists a large number of unlabeled users, and it is expensive to obtain their identities. Existing research efforts on shilling attack detection fail to exploit these unlabeled users. In this article, *Semi-SAD*, a new semi-supervised learning based shilling attack detection algorithm is proposed. Semi-SAD is trained with the labeled and unlabeled user profiles using the combination of naïve Bayes classifier and EM-λ, augmented Expectation Maximization (EM). Experiments on MovieLens datasets show that our proposed Semi-SAD is efficient and effective.

Categories and Subject Descriptors: H.3.3 [Information Search-Retrieval]: Information Filtering

General Terms: Performance, Security

Keywords: semi-supervised learning, shilling attack detection, naïve Bayes,EM

1. INTRODUCTION

Since the filtering process of Collaborative filtering (CF) is based on the profiles of other users, the CF based recommender systems are vulnerable to *shilling attacks*. In the recommender systems applied to e-business, suppliers have natural motivation to utilize the shilling attack to promote their products. Therefore, how to detect the shilling attacks is a big challenge in the studies of recommender systems.

Some leading e-commerce web sites, e.g. Amazon and Taobao, have quite many users but only a small number of their users can be obviously identified as normal users or shilling attackers. For instance, in Taobao, crown shop owners and buyers with high positive ratings can be identified as normal users, and the users with fraud history or very low positive ratings can be identified as shilling attackers. While the natures of other users with moderate positive ratings cannot be identified without manual analysis. The identified users are called *labeled user*, and others are *unlabeled user*.

Most existing shilling detection algorithms are based on supervised learning such as [1-3]. These supervised algorithms need a large number of labeled users to enhance the accuracy. An unsupervised shilling attack detection algorithms using *principal component analysis* (PCA) is proposed in [4]. Hurley et al. utilize Neyman-Pearson theory to construct both supervised and unsupervised detector [5]. It is obvious that unsupervised learning based detectors do not make good use of the labeled data which indeed exists and is significant for the detection.

In our work semi-supervised learning [6], a kind of machine learning (ML) technique, is employed to exploit unlabeled users in addition to labeled users for shilling attack detection. Shilling attack detection is a binary classification problem involving two classes. Semi-supervised learning based Shilling Attack Detection (Semi-SAD) algorithm is proposed in this paper. For convenience, Semi-SAD utilizes naïve Bayes as the initial classifier and Expectation Maximization (EM) to improve the classifier.

2. RELATED WORK

From the intention perspective, shilling attacks can be divided into *push* and *nuke* attacks to make a target item more or less likely to be recommended respectively. Let I_s denote the selected item set, I_F denote filler item set, and i_t denote target item. Five shilling attack models are introduced as follows:

- **Random attack:** $I_s = \emptyset$; give I_F random ratings; $i_t = r_{max}$;
- **Average attack:** $I_s = \emptyset$; the ratings for I_F are distributed around the mean for each item i; $i_t = r_{max}$;
- **Segmented attack:** I_s are the target item's similar items and $I_s = r_{max}$; $I_F = r_{min}$; $i_t = r_{max}$;
- **Bandwagon attack:** I_s are the frequently rated items and $I_s = r_{max}$; give I_F random ratings; $i_t = r_{max}$;
- **Sampling attack:** $I_s = \emptyset$; copy a existing user profile as I_F; $i_t = r_{max}$.

Williams et al. proposed three obfuscated techniques: *noise injection*, *user shifting* and *target shifting* [2]. Since user shifting is the general expression of noise injection, noise injection and target shifting are adopted in this paper to obfuscate the attack profiles. Hurley et al. presented a

RecSys'11, October 23–27, 2011, Chicago, Illinois, USA.

UID/Profile	Class S/N	Metric 1 *H(X)*	Metric 2 *ADegSim*	.....	Metric n *RDMA*

Figure 1: Data format after pre-processing

simple and effective strategy to obfuscate the average attack, *Average over Popular items* (AoP) [5]. AoP x% attack chooses filler items with equal probability from the top x% of most popular items, rather than from the entire catalogue of items. It is reported that PCA-based detector is ineffective to AoP attack. We compare Semi-SAD with a PCA-based detector against AoP attack in this paper.

3. PROBLEM STATEMENT

In order to detect the shilling attackers among the sea of user profiles, pre-process is performed on the user profiles to format the data as shown in Fig.1 UID uniquely identifies a user and is the main key of a profile. The value of class is *Shilling (S)* or *Normal (N)*. For training data, the class value is labeled. For testing data, the class value is unknown and the detection algorithms aim to determine the class of the user. Metrics are used to describe the characteristics of the normal users or the shilling attackers. No single metric can distinguish the normal users from the shilling attackers entirely. Therefore, we should select and define a series of metrics. In this paper five metrics are selected among which *entropy* is newly defined and *DegSim*, *LengthVar*, *RDMA*, *FMTD* stem from [2].

Definition 1. Let $X_u = n_i, i = 1, 2, ..., r_{max}$ be an statistical set of user u, where n_i means the occurring time of the i-th possible rating in the u's profile. The entropy $H(u)$ is computed by Eq.(1).

$$H(u) = -\sum_{i=1}^{r_{max}} \frac{n_i}{S} log_2 \frac{n_i}{S}, \quad where \;\; S = \sum_{i=1}^{r_{max}} n_i \qquad (1)$$

Entropy measures the degree of dispersal or concentration of user profiles. The value of $H(u)$ is in the range [0,$log_2 r_{max}$]. The minimum value 0 is taken when all ratings are the same ($r_{max} = 1$) and the maximum value $log_2 r_{max}$ is taken when $n_i = n_j$ for all *i,j*.

Next, we present the formal statement for shilling attack detection problem. The original input is the set of rating records of all users. Let L denote the labeled user profile set with size $|L|$, $L = (u_1, c_1), (u_2, c_2), \cdots, (u_{|L|}, c_{|L|})$, and U denote unlabeled user profile set with size $|U|$, $U = u'_1, u'_2, \cdots, u'_{|U|}$. If the category of a user (shilling or normal) is known, the user belongs to L, otherwise the user belongs to U. There is a one-to-one correspondence between any user in U and category.

The shilling attack detection algorithms try to learn the function $f : U \rightarrow C$ in order to accurately predict the class label c for any user profile u. The class label set is written $C = \{S, N\}$. u_i, $u'_j \in U$ are the UID as shown in Fig.1 $c_i \in C$ is the class label categorized into *normal(N)* and *shilling(S)*.

4. SEMI-SAD ALGORITHM

The proposed the Semi-SAD algorithm consists of two phases. The first phase trains a naïve Bayes classifier on a small set of labeled data and the second phase incorporates unlabeled data with EM-λ to improve the initial naïve Bayes classifier.

4.1 Phase 1: Training a Naïve Bayes Classifier on Labeled Data

The naïve Bayes classifier, a well-known probabilistic classifier, is employed to estimate the probability of an unknown user profile belonging to a certain class. We assume the i-th metric M_i follows the Gaussian probability distribution with mean μ_i and standard deviation σ_i. $P(x_i|C)$ is the probability of a user profile satisfying that its i-th metric $M_i = x_i$ and it belongs to the class C.

$$P(x_i|C) = g(x_i, \mu_{Ci}, \sigma_{Ci}),$$
$$where \qquad g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (2)$$

μ_{Ci} and σ_{Ci} are the mean and standard deviation of the i-th metric of the labeled data in class C. Then, for the pre-processed user profile u in the data format of Fig.1, the probability of u belonging to the class C is given by Eq. (3).

$$P(u|C) = \prod_{i=1}^{n} P(x_{ui}|C) \qquad (3)$$

With Eqs.(2) and (3), we can calculate the probability of an unknown user profile belonging to a certain class. As a matter of fact, the labeled data is utilized to determine the parameters, i.e. mean and standard deviation, of probability distribution of each class.

4.2 Phase 2: Incorporating Unlabeled Data with EM-λ to Improve the Classifier

The advantage of semi-supervised learning method over supervised learning method is that the unlabeled data can be exploited to improve the classifier. Especially when the labeled data is sparse, the parameter variance from the naïve Bayes classifier is so big that the accuracy of shilling attack detection is unacceptable.

EM algorithm is a widely used approach to exploit the unlabeled data. EM-λ, an augmented EM proposed in [6], modulates the influence of the unlabeled data by adding a weighting factor λ in the estimation. It iteratively re-estimates parameters by repeating its two kinds of steps (E-step and M-step) until converging to a stationary value for the estimation. The estimated parameters in shilling attack detection are the mean and standard deviation of both normal class and shilling class.

- E-step: Calculate the probability of each user profile belonging to a class from Eq.(4).

$$P(u_k \in C) = P(C|u_k) = \frac{P(C)P(u_k|C)}{P(u_k)} \qquad (4)$$

  where $P(u_k)$ is constant, and we assume $P(S) = P(N)$ for unknown data. Therefore, $P(u_k \in C)$ is only determined by $P(u_k|C)$ that can be obtained from Eqs.(2) and (3).

- M-step: Calculate the estimated parameters based on the probability calculated in E-step. The mean of the

i-th metric on the data belonging to class C can be computed as:

$$\mu_{Ci} = \frac{1}{|C|}\sum_{u=1}^{|C|}\omega_u x_{ui} \tag{5}$$

The standard deviation of the i-th metric on the data belonging to class C can be computed as:

$$\sigma_{Ci} = \sqrt{\frac{1}{|C|}\sum_{u=1}^{|C|}\omega_u^2(x_{ui}-\mu_{Ci})^2} \tag{6}$$

Note that in Eqs.(5) and (6), the number of each class $|C|$ is also computed by adding weight ω_u. $|C|$ represents $|N|$ and $|S|$ computed by Eq.(7).

$$|C| = \sum_{u=1}^{|L|+|U|}\omega_u \tag{7}$$

The probability of a user belonging to a certain class (N or S), is employed as a weighting factor in the estimation. In Eqs.(5), (6) and (7), ω_u can be computed by Eq.(8):

$$\omega_u = P(C|u_k) = \frac{P(u \in C)}{\sum_j P(u \in C_j)} \tag{8}$$

EM-λ has the same E-step as EM, but the M-step is different with the following entail for Eq.(8). First $\Lambda(u)$ is defined to be the weighting factor if u is in the unlabeled set, and 1 if u is in the labeled set:

$$\Lambda(u) = \begin{cases} \lambda, & if \quad u \in U \\ 1, & if \quad u \in L \end{cases} \tag{9}$$

Then, we can rewritten Eq.(8) as:

$$\omega_u = P(C|u_k) = \Lambda(u)\frac{P(u \in C)}{\sum_j P(u \in C_j)} \tag{10}$$

It is obvious that when λ is close to zero, the unlabeled data will have little influence to the shape of EM's hill-climbing surface. When λ =1, each unknown user profile will be weighted as known user profiles, and EM-λ squints towards to the original EM algorithm.

4.3 Pseudo-code of Semi-SAD

The Semi-SAD algorithm is designed to integrate the naïve Bayes classifier presented in section 4.1 with the EM-λ presented in section 4.2. The pseudo-code of the Semi-SAD is presented in Table 1.

Semi-SAD starts with the labeled user profile set, L, and the unlabeled user profile set, U. The user profile is the user's rating for some items, for example, one user rates movies he/she watched in the MovieLens dataset. The naïve Bayes classifier are trained on the labeled user profile set, L, as the initial classifier. Then, Semi-SAD starts the iteration of E- and M-steps until the estimated parameters become stable. In practice, we determine that the convergence has occurred if observing a below-threshold change of Eqs.(5) and (6). Once the iteration stops, the initial classifier has been improved by EM-λ, and returned it as the shilling attack detector. For any unknown user profile, the shilling attack detector can predict its class label.

5. EXPERIMENTAL RESULTS

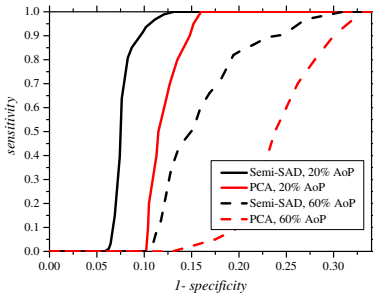

Figure 2: Semi-SAD vs. PCA against AoP attack

The MovieLens datasets published by GroupLens [7] are used in the experiments. This dataset consists of 100,000 ratings on 1682 movies by 943 users. All ratings are integer values between one and five where one is the lowest (disliked) and five is the highest (most liked). Various shilling attack profiles are generated and injected into MovieLens datasets. We define two staple metrics *specificity* and *sensitivity* to evaluate the performance of the Semi-SAD.

$$sensitivity = \frac{\#truepositives}{\#truepositives + \#falsenegatives} \tag{11}$$

$$specificity = \frac{\#truenegatives}{\#truenegatives + \#falsepositives} \tag{12}$$

The *specificity* measures the proportion of correctly identified normal profiles, and the *sensitivity* measures the proportion of correctly detected attack profiles.

5.1 Performance of Semi-SAD

The AoP attack introduced in [5] is a simple and effective obfuscated average attack. In this section, we compare the anti-AoP performance of Semi-SAD with PCA-SAD. ROC curves are employed to describe the performance of two detectors. ROC space is defined by *1-specificity* and *sensitivity* as x and y axes respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs).

Semi-SAD decides the class label of the k-th profile by comparing the ratio of $P(u_k \in S)$ to $P(u_k \in N)$ to a threshold η. Therefore, For Semi-SAD, the ROC of Semi-SAD is generated by varying the threshold η. For PCA-SAD, the ROC is generated by varying the attack cluster size. Fig.2 illustrates the ROC curves of two detectors in AoP 20% attack and AoP 60% attack respectively. The results show that Semi-SAD has higher probability of good detection and lower probability of false alarm than PCA-SAD.

5.2 Impact of λ

The experiments in this section investigate the impact of weight λ in the second phase of Semi-SAD. We range λ from 0 to 1 and the interval is set to 0.1.The size of labeled data is also considered in the experiments. Three scales of labeled data size are chosen: $50N - 25S$, $100N - 50S$, $150N - 75S$. AoP attack is tested, and the filler size is set to 5%. Shilling profiles in training dataset are AoP 30% attacks, and the *actual* attacks are AoP 60% attacks.

Fig.3 and Fig.4 show that *sensitivity* and *specificity* of

Table 1: Pseudo-code Describing the Semi-SAD algorithm

Input: L: Labeled user profile set, U: Unlabeled user profile set
Output: A shilling attack detector, θ, that takes an unknown user profile and predicts a class label.
1: Take pre-process on L and U to derive the data format shown in Fig.1.
2: Train an initial naïve Bayes classifier, θ, based on labeled user profile set, L, only.
3: **repeat until** none of estimated parameters (μ_{Ci} and σ_{Ci}) changes
4: (E-step) Utilize the current classifier, θ, to calculate $P(u_k \in C)$ by Eq.(4).
5: (M-step) Improve the current classifier, θ, by utilizing Eqs.(5) to (10).
6: **end of repeat**
7: Return the classifier, θ, as the shilling attack detector.

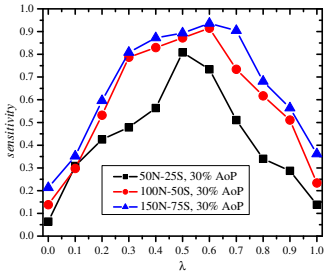

Figure 3: Impact of λ on sensitivity

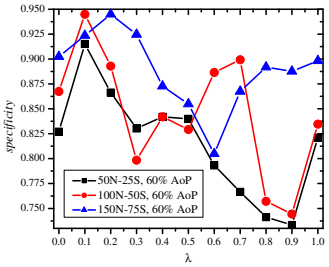

Figure 4: Impact of λ on specificity

Semi-SAD varies with the increase of λ in three scales of labeled data size respectively. Three curves in Fig.3 exhibit similar variation tendency. The curves reach a peak when the λ value is intermediate. Meanwhile, the curve of small labeled data size changes more significantly than that of large labeled data size. However, as the λ value increases, the *specificity* of three different labeled data size varies irregularly. It arises because the normal user profiles in training dataset and test dataset are similar. However, *specificity* does not benefit from the weight adjustment of unlabeled data as in Fig.4 Because the labeled shilling profiles are very useful to guide parameter estimation of EM, the smaller labeled data size is, the smaller weight unlabeled data should be given.

Experimental results shown in Fig.3 and Fig.4 indicate that when the labeled data size is very small, carefully selecting λ could significantly improve the practical performance of shilling detection even further. Therefore, we can utilize *cross-validation* approach to determine the optimal λ value for the specific data.

6. CONCLUSION

It is observed that there are few labeled users and a great deal of unlabeled users in the real recommender systems. Existing research efforts on shilling attack detection are based on either labeled data, or unlabeled data. In this article, we present a novel semi-supervised learning based shilling attack detection algorithm, *Semi-SAD*. It learns from both labeled and unlabeled user profiles based on the combination of a naïve Bayes classifier and EM-λ. Experiments on MovieLens demonstrate that Semi-SAD has a great advantage over a PCA-based detector, and the performance of Semi-SAD could be improved significantly if we carefully select λ, especially for small labeled datasets.

7. ACKNOWLEDGMENTS

This research is supported by National Natural Science Foundation of China under Grants No.71072172, the program for New Century Excellent Talents in university under Grants No.NCET-07-0411 and Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201.

8. REFERENCES

[1] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik. Classification features for attack detection in collaborative recommender systems. Proceedings of the 12th ACM SIGKDD, New York, USA, 2006.

[2] C. Williams and B. Mobasher. Profile Injection Attack Detection for Securing Collaborative Recommender Systems. DePaul University CTI Technical Report, 2009.

[3] B. Mobasher, R. Burke, C. Williams, and R. Bhaumik. Analysis and detection of segment-focused attacks against collaborative recommendation. Proceedings of the 2005 WebKDD Workshop, 2005.

[4] B. Mehta and W. Nejdl. Unsupervised strategies for shilling detection and robust collaborative filtering. User Modeling and User-Adapted Interaction, vol. 19, No. 1, 2009, pp. 65-97.

[5] N. Hurley, Z. Cheng, and M. Zhang. Statistical attack detection. Proceedings of ACM Conference on Recommender Systems (RecSys '09), New York, USA, 2009.

[6] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. Mach Learn, vol. 39, No. 2, pp. 103-134, 2000.

[7] GroupLens Research. http://www.grouplens.org/node/73, 2011.