International Journal of Security and Its Applications
Vol.8, No.4 (2014), pp.253-264
http://dx.doi.org/10.14257/ijsia.2014.8.4.23

Robust Online Filter Recommended Algorithm based on Attack Profile

Gao Feng

College of computer, Changchun Normal University,
Jilin 130032, China
Gaofengg2014@126.com

Abstract

In view of the high vulnerability of traditional user-based recommendation algorithm to shilling attacks, In this paper, on the basis of the work of the group effect on the attack profiles, this paper analyzes the statistical features of the nearest neighbors of target users before and after attack, Design a kind of Attack Profiles online filter to attack the target user profile from the nearest neighbor filter. And this filter improves the user-based recommendation algorithm nearest neighbor selection strategy, thus proposes the Collaborative Recommendation algorithm based on Online Filter for Attack Profiles (CROFAP). Experiments show that attack profile online filter can accurately identify and filter out most attacks profile to ensure the robustness of the CROFAP algorithm.

Keywords: *Recommender system, shilling attack, robust recommendation, shilling attack detection*

1. Introduction

This paper is highly sensitive for the traditional user-based recommendation algorithm to shilling attacks, Committed to the development storage robust recommendation algorithm. This algorithm adopt the more reliable the nearest neighbors selection strategy. To weaken the adverse effect of attackers to score prediction [1-2].

Integrating Shilling attack detection technologies with the recommendation algorithm is to enhance effective way of the recommendation system robustness. At present, there are mainly two kinds of integration strategy. One is using the shilling attack detection pretreatment score matrix, namely in off-line way to remove all attackers, and then run the recommendation algorithm. The other one is shilling attack detection technology into organic recommendation algorithm, Allow the real users and attackers coexist in the system, only for the score in the prediction stage is online detecting and shielding the attacker. The first strategy is simple and intuitive, and can be directly used the existing recommendation algorithm and unchanged. But in view of the existing shilling attack detection technology in unsupervised problems and universality lacking [3]. Inevitably there will be leakage delete attackers even deleted real user questions, this will have the authenticity or integrity loss recommendation algorithm of data source, thus unable to ensure the accuracy and robustness of the recommendation algorithm. Second strategies need to modify or redesign recommendation algorithm. This is not the first strategy simple, but because the score predictions depend only on local data (such as nearest neighbor) rather than the entire score matrix. It is easy to take some heuristics to design efficiently shilling attack online detection method. Effect of shilling attack defense is ideal. To sum up the above, this paper adopt second strategies, proposed the collaborative

ISSN: 1738-9976 IJSIA

recommendation algorithm based on attack profile online filter (CROFAP). Experimental results show that this online filter can effectively filter out attackers of nearest neighbors [4].

2. Recommendation Algorithm based on User

Recommendation algorithm based on user is a simple, effective, easy to deploy recommendation algorithm [5-6]. Widely used in recommender systems. The operation principle of this algorithm to simulate the process of daily life in between the relatives and friends are interested in passing mouth to mouth recommended procedure. Figure 1 can visualize this process. First, the central target users find the user group in the nearest neighbor. It is in the circle of three users. Need to pay attention to user distance measure is no longer a general sense of the spatial distance, but is the degree of similarity of interest. Secondly, the nearest neighbor after interaction will recommend their favorite things to target users. Given the highly interest similarity, the target users will great Favorite these recommended information [7].

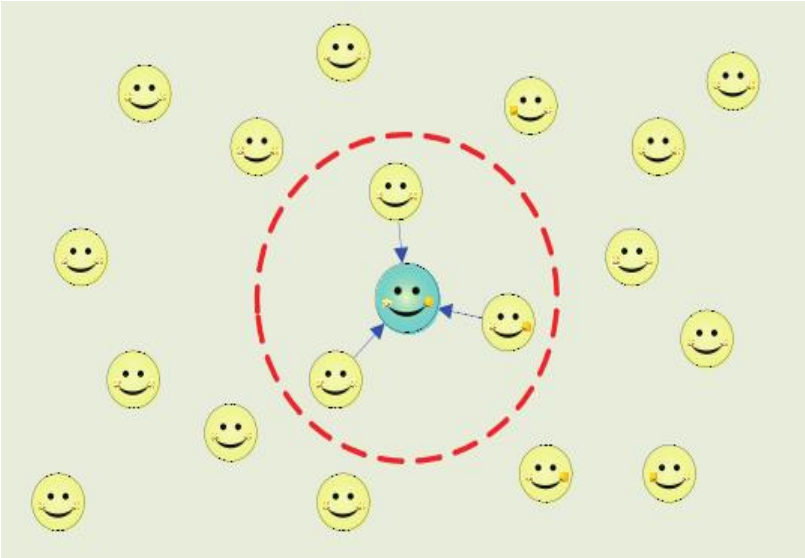

Figure 1. The Recommendation Algorithm based on User's Operation Principle

Recommendation algorithm based on user is using the principle of predictive target user loss score, and the high evaluation item recommends targeting users. Formally, the algorithm to predict the target $user_t$ prediction $item_j$ scoring process can be divided into three steps:

Calculation $user_t$ interest similarity of all users in system, Similarity measure selected Pearson correlation in most cases. The Pearson correlation between $user_t$ and $user_n$ was acquired by formula (1), which CR_m represents common evaluation of two users.

$$sim(user_t, user_n) = \frac{\sum_{item \in CR_m} (R(user_t, item) - \overline{R}_{user_t})(R(user_n, item) - \overline{R}_{user_n})}{\sqrt{(R(user_t, item) - \overline{R}_{user_t})^2} \sqrt{(R(user_n, item) - \overline{R}_{user_n})^2}} \quad (1)$$

From all the users of the evaluation $item_j$ .Select the K with the highest similarity of $user_t$ r as the nearest neighbor $user_t$, thus set to KNN.

By $user_t$ nearest neighbor, according to the formula (2) weighted scoring method to predict $item_j$ score.

$$predict(user_t, item_j) = \bar{R}_{user_t} + \frac{\sum_{user \in knn} sim(user_t, user)(R(user, item_j - \bar{R}_{user})}{\sum_{user \in knn} sim(user_t, user)} \quad (2)$$

The score prediction is based on the K nearest neighbors of target user, so the recommendation algorithm based on user is also known as K nearest neighbors (K-NN) algorithm. In general, at the same time specified similarity nearest neighbor and target users should be greater than a certain threshold φ , Because of the similar spending low or even negative neighbor is apparently useless on the accuracy rating prediction. So, according to the prediction of the target user, nearest neighbor sometimes will be less than K .

The recommendation algorithm based on user is simple and effective, but its robustness is not ideal. The experimental results prove that the algorithm is highly sensitive to shilling attack; a small amount of attack profile can significantly change the recommendation result. Algorithm of Section (2) step is the root of problem of this robust. Because the algorithm on the selection of the nearest neighbor is only consider the similarity, but other available information does not refer to user. Attacker exploits this vulnerability effectively to improve their similarity with real users through a variety of ways. To construct the random attack, man attack attacks and popular way to build shilling attacks despite the different. But try to improve the similarity of attackers and most real users for the purpose. So you can increase the probability of attack profile appears in the nearest neighbor, to the target user recommendation influence. In extreme cases, all nearest neighbor constituted by attack profile, The mean attack to attack the best is as an example, the argument is $p^{att} = 10\%$, $p^{fill} = 3\%$ the mean attack inject MovieLens100K datasets, the set of nearest neighbor number is $K = 30$, the thresholds is $\varphi = 0.1$. Figure 2 shows an overview of attack profile in abundance of the nearest neighbor. As can be seen, with the target user vary certain fluctuations, despite the abundance of attack profile, but on the whole attack profile in the nearest neighbor still occupies the important proportion. Through the further experiment, observed similar phenomena exist in random attack and popular attack. Obviously, in order to reduce algorithm the sensitivity of shilling attack, an intuitive solution is to remove attack profile in nearest neighbor

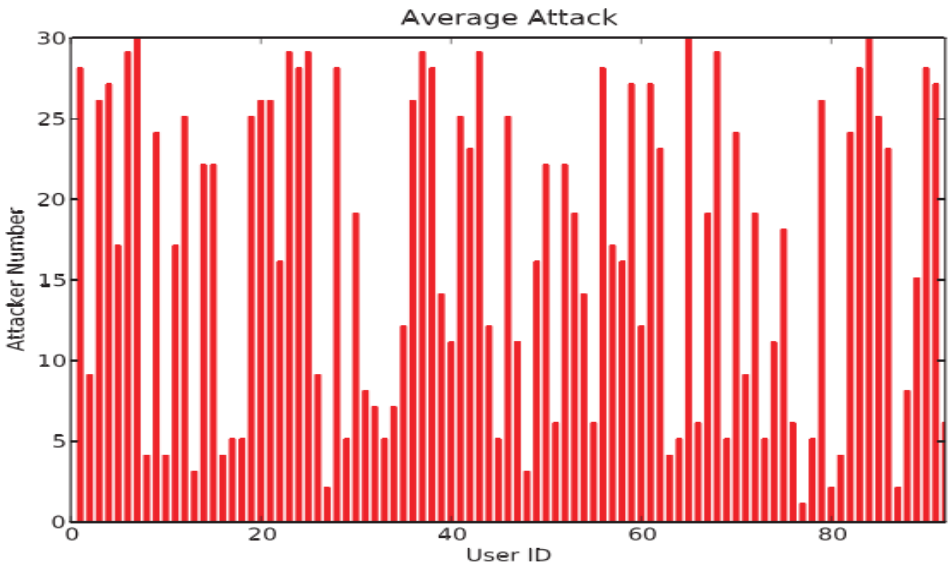

Figure 2. Attack Profile Abundance of Nearest Neighbor

3 Attack Profile Online Filter

To try to prevent attacks overview into the nearest neighbor, CROFAP algorithm in this paper follows the recommendation algorithm based on the user's basic framework, and with the help of attack profile online filter, improved the original algorithm step2 nearest neighbor selection strategy.

3.1. Before and After Target of Shilling Attacks in Statistical Characteristics of User's Nearest Neighbor

The nearest to the target $user_t$ matrix $X_t = [u_1, u_2, ..., u_K]^T$, Definition:

$$Advo(X_t) = \sqrt[2K]{\det(X_t X_t^T}$$

For the nearest neighbor of any target user, the $Advo$ value corresponding to the user with the attack profile in the nearest neighbor proportion increases. This means that the A $Advo$ value can be used as important evidence indicating the existence of the attack profile. The $Advo$ value is more; the existence of the attack profile in the nearest neighbor is the greater. On the contrary, the possibility is the smaller.

However, the current value to determine whether the mere existence of $Advo$ profile attacks nearest neighbor is not rigorous. With a large numerical and at the request of $Advo$, the nearest neighbor number is not less than a certain limit. Although the default number of nearest neighbors K can be approximated to ensure the general theorem, the similarity threshold of φ presence may lead to the target user cannot take sufficient K nearest neighbor. So, once the nearest neighbor is less and mostly by interest real user unique composition, it will have a larger $Advo$ value. The extreme case, if the nearest neighbor is only one user profile, regardless of whether the user is true, $Advo$ will be taken to a maximum is 1. This requires that we depend on the actual number of nearest neighbors original $Advo$ be amended to increase its presence as evidence of the credibility of the attack profile. Accordingly, this paper defines the following modified $Advo$ function:

$$Advo(X_t) = Advo(X_t)^{\frac{rows(X_t)}{K}}$$

Among, the function $rows(X_t)$ returns the number of rows of the matrix X_t, $user_t$ is also the actual nearest neighbor number. Apparently, $rows(X_t) \in \{1, 2, ..., K\}$, When the actual nearest neighbor number is less than preset value of K. $Advo$ original value is multiplied by the corresponding penalty coefficient, to some extent to be present he water.

Intuitively, the data set is now MovieLens100K injected the mean attack of $p^{att} = 10\%$, $p^{fill} = 3\%$ Figure 3 shows the nearest neighbor $Advo$ distribution of attack target user of before and after. In Figure 3, the vertical axis is the data point added random perturbations. When there is attack profile in the nearest neighbor, most of the data points to attack the data points on the right side, the corresponding $Advo$ value has been significantly improved. This phenomenon when does not exist the attack, the nearest neighbor preferences are similar to the target user. So the nearest higher interest coincidence degree determines $Advo$ value is not too high. To be floating in the lower range. Once is the attack profile into the nearest neighbor, attack profile points of interest of extreme dispersion will increase $Advo$ .In summary, $Advo$ can to a certain extent to reveals the existence of attack profile. Further experiments showed that before and after the random attacks and popular attack also has a similar situation. This inspires us to reduce the nearest $Advo$ value of user as the goal, selectively delete user profile, thereby indirectly to filter out attack general purpose.

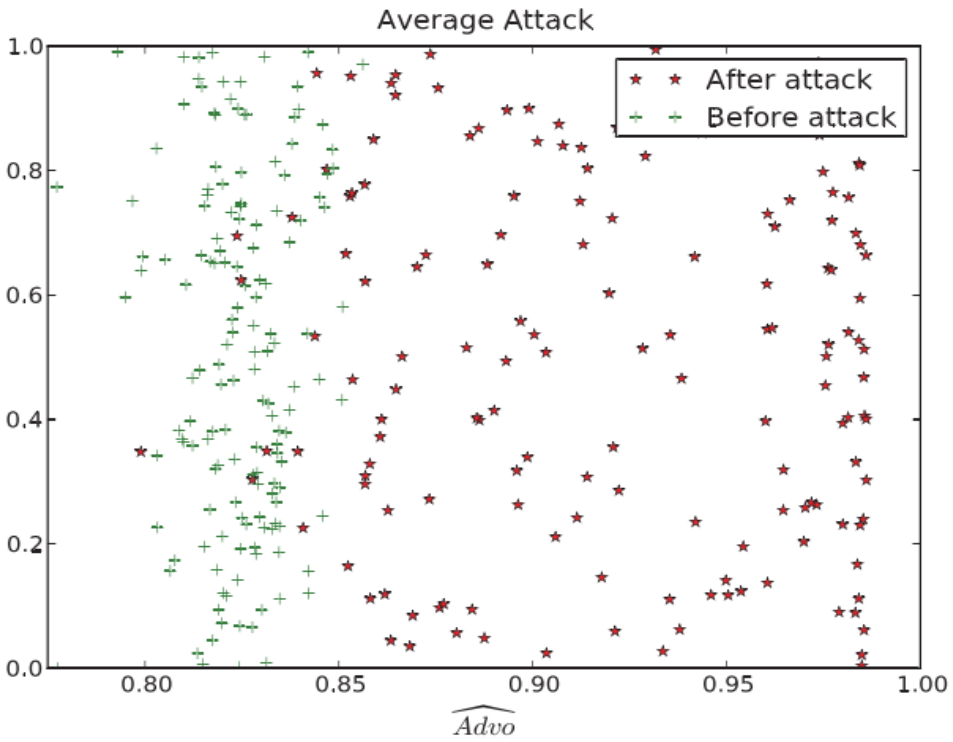

Figure 3. Distribution of the Nearest Neighbor $Advo$ Value Before and After of the Attack

3.2. The Filter Process and Interpretation

First, given the filter process:

Input: Score $item$ to be evaluated user sequences $userlist$, where, the user in accordance with the target $user_t$ similarity is descending order.

Output: $user_t$ neighbor list is KNN

Step 1: $count \leftarrow 0$

Step 2: $KNN \leftarrow userlist[1:K]$

Step 3: Looking for a user $user^*$ from KNN, meet the conditions:

(a). $R(user^*, ltem) = r_{\max} or\ r_{\min}$

(b) $user^* = \arg\min_{user} Advo(KNN - user)$, which, $Advo(KNN - user)$ is except $user$ remaining user profile corresponding $Advo$ value in the KNN, $user$ select from satisfying the condition (a) users.

Step4: If $user^*$ does not exist, the terminating is execution, returns KNN; otherwise continue.

Step5: Generate $userlist'$ a temporary copy of $userlist$, and deleting $user^*$ from $userlist'$ $KNN \leftarrow userlist'[1:K]$

Step6: If $Advo(KNN') > Advo(KNN)$, then is count +1, otherwise is $count \leftarrow 0$.

Step7: With probability $(1 - \exp(-\frac{count}{10}))^a$ is termination of execution, Returns KNN; otherwise $user^*$ is removed from $userlist$, go to Step 2

The above workflow Step 3, Step 6 and Step 7 is a key point. Step 3 will condition (a) (b) as the sufficient conditions for judging. By scanning the KNN list, find and meet the conditions of the users, thus considers that this user strongest attack suspects. The two conditions is established according to certain facts. Condition (a) adopts heuristic judgment strategies. If the score of $item$ is exactly the goal of shilling attack, then with few exceptions attack profile $item$ is bound to get extreme score, to some extent can be considered to play extreme for $item$ is rated attacker. Condition (b) user selection strategy is similar to the greedy method. The program is chosen satisfies the condition (a) users from KNN, and calculating their residual $Advo$. Determination $Advo$ user of minimal residual is the suspected attacker. Because the number of fixed user profile, $Advo$ is the smaller, and the possibility of attack profile containing general is also lower. So with minimal residual $Advo$ user is most likely the attacker. However, the user simply identifies is only the suspected attacker, it cannot ventured deleted. Step 6 is to judge the nearest $Advo$ delete this user is greater than before delete. If more than, that is likely to accidentally deleted the real user, but at the same time, it cannot be completely ruled out the possibility that the right to delete the attacker. It was not a time to terminate the program, it will only counter+1. Otherwise, that the right to delete the attacker, the counter is reset. Step 7 adopts the probability of termination of the execution of the program counter. Obviously, count is increased, Termination $(1 - \exp(-count/10))^a$ is greater. Especially when there is a continuous increase in the value of the nearest neighbor $Advo$ delete operation, it means that the attack may have been deleted exhausted overview, thus the possibility of real users mistakenly deleted surge, then a larger count values will make ensure the timely termination of the

program. The actual effect of attack profile filter will show in the experimental part. It should be emphasized; the pre-filter operation can be computed covariance matrix of the whole system composed of the user profile. Thus, when a user is seeking $Advo$ subset profile, it can be read directly the corresponding sub-matrix. From the precomputed covariance matrix, without double counting, do not need to repeat the calculation. Thus adopting the attack profile filter to replace based on user the recommendation algorithm Step 2. The remaining two-step is unchanged, that is proposed CROFAP algorithm.

3. Experimental Analysis and Results

4.1. Data Sets and Experimental Setup

The experimental data sets MovieLens100K and MovieLens1M. For the MovieLens100K data set, the experimental adopts all users and items. For the Movie Lens1M data set, to adopt all the items, but chosen randomly 1/4 (about 1510) users. Experimental training set and test sets defined as the ratio of 9: 1. All the experiments of fixed threshold is φ = 0.1.

In order to validate the robustness of the CROFAP algorithm, assume that the data from original users for real users.

In order to validate the robustness of the CROFAP algorithm, assume that the data from original users is real users. In different attack strength p^{att} and fill rate p^{fill}, the data set into three kinds of attacks: random attack, mean attack and popular attack, which, the popular attack filling selection is popular the highest degree of individual items. Robust quantitative measure used two indicators: the mean absolute error (MAE) and prediction error (PS). Calculate MAE were retained for each test users for prediction.

4.2. The Filtering Effect of Attack Profile

First, define the amount of $filtered$ and accuracy of $precision$, $F^{all}(user)$ is filter from the user neighbor list to filter the user profile set. $F^{att}(user)$ Is collection of attack profile in $F^{all}(user)$. For the target user set (the test set) is:

$$\begin{cases} filtered = \dfrac{\sum_{user \in U_t} |F^{all}(user)|}{|U_t|} \\ precision = \dfrac{\sum_{user \in U_t} |F^{att}(user)| / |F^{all}(user)|}{|U_t|} \end{cases}$$

Apparently, $filtered$ is from each of the neighbors of the target user list to filter average number of the user profile. $precision$ Is the average proportion of attack profile in the filtering user profile?

Now MovieLens100K data set Injection into $p^{att} = 10\%$, $p^{fill} = 3\%$ set is K=30. Figure 4 shows the work effect of the filter. Compared with the previous filter, in the nearest neighbors abundance of the attack profile is decreased. Although take different a, filter capacity will be differences. But the accuracy is above 70%, which indicates that the filter has strong recognition ability. Most of filtering user is to attack profile.

As a increases, the abundance of attack profile also show a downward trend, but this does not mean a bigger is better. Increased a will reduce the filter termination probability; it will cause a large amount of filter. Figure 4 is this phenomenon can be observed. At the same time as the accuracy slightly decreased, so the amount of filtering more will cause more real user profile of the error filter, it is not conducive to the accuracy of the algorithm.

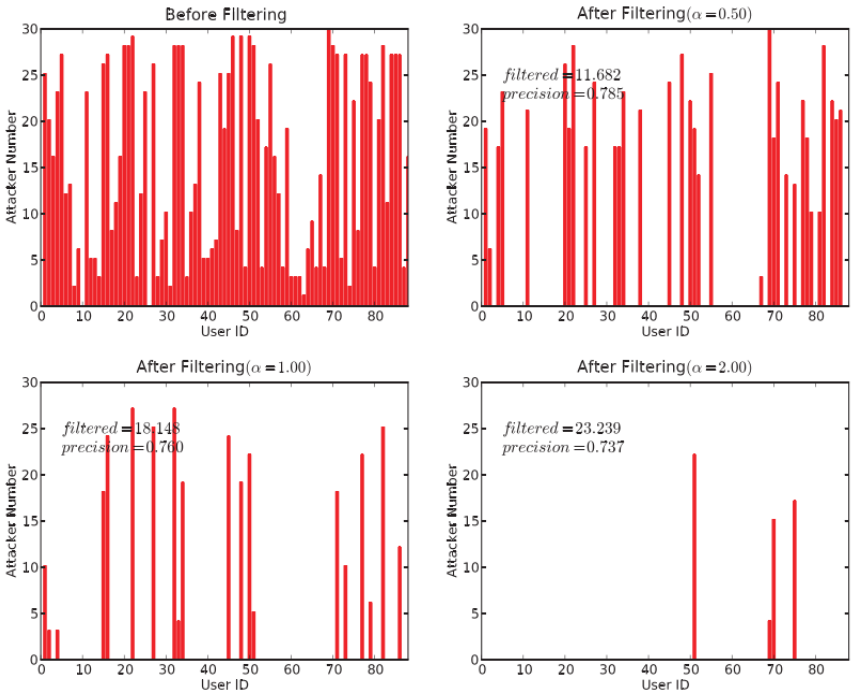

Figure 4. The Filtering Effect of Attack Profile

4.3. Parameter Selection

The nearest neighbor number K and of a termination probability are two key parameters of CROFAP algorithm. By examining the two parameters affection of the robustness and the experimental results to determine the parameter values.

First, the K should not be too small or too large. K is too small without include nearest neighbor, it will reduce the accuracy of prediction. Too large increases the computational burden. On the one hand will attract more users to the low correlation, not conducive to further improve the accuracy, and even damage the accuracy. In particular, the similarity threshold are also too much K values to make meaningless. For example, fixed a is 0.5, figure5 shows the effect of K value on MAE. In the MovieLens100K data. MAE decreased with increasing K. But when $K > 30$ MAE has no obvious improvement, so there is no need to further increase the K value. And in the MovieLens1M data, MAE reached to increase after the lowest value when $K = 40$, it should not continue to increase the K value. This showed that K should be chosen in the range of a moderate.

Secondly, a is the larger, the filter termination probability is smaller, to filter out attack profile is more also, this means that the PS should also be reduced. Figure 6 confirms this judgment. In this case the data set exists the average attack of $p^{att} = 5\%$ $p^{fill} = 3\%$. And fixed K is 30. But a is not the bigger the better, because MAE increases with a upward

trend. This is because the filter is filtered off while the attack profile, it is deleted by a certain amount of real user profile. But a is the larger, real user profile filter is more also, this will reduce the algorithm accuracy. So a value should according to selection of the actual situation. In summary, combined with experimental results. For MovieLens100K dataset, set the parameter K = 30, a = 1.6. For MovieLens1M dataset, set the parameter K = 40, a = 1.6.

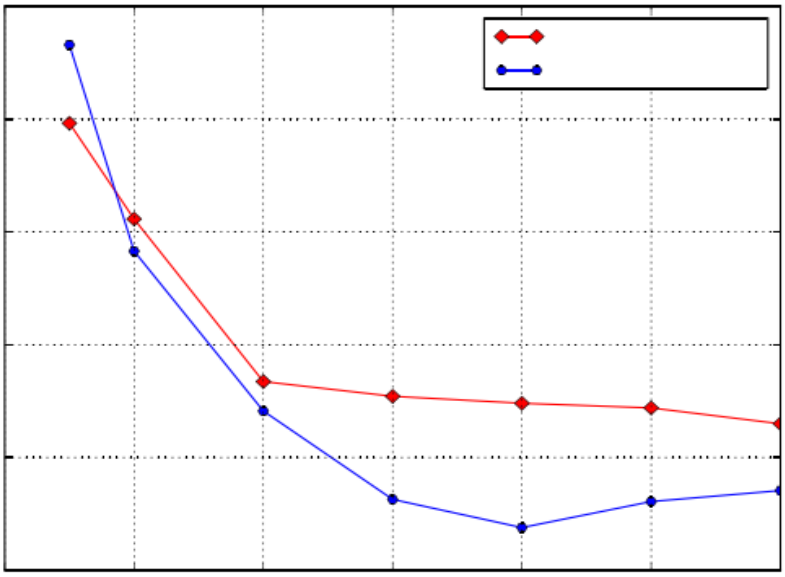

Figure 5. Influence of Parameter K for MAE

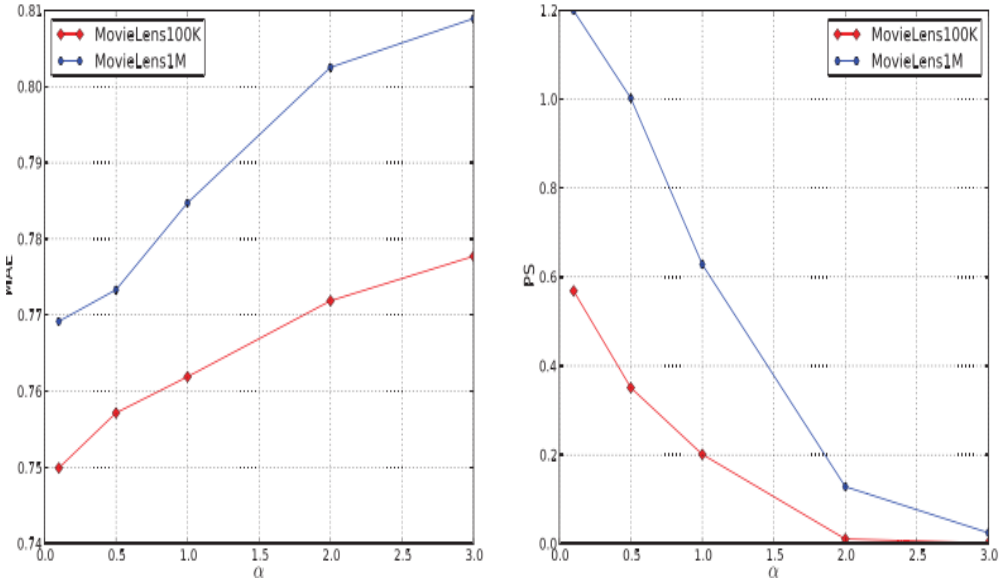

Figure 5. The Influence of Parameter a for MAE and PS

4.4. The Experimental Results

First, a comprehensive assessment of the robustness of the CROFAP algorithm in the MovieLens100K and MovieLens1M data set. The overall Assess the robustness of CROFAP

algorithm. The experiment adopted 3 * 4 * 5 design patterns (random attack, mean attack, popular attack), the attack strength of $p^{att}(5\%,10\%,15\%,20\%)$ and filling rate of $p^{fill}(3\%,6\%,12\%,15\%)$ corresponds to a different combination of a group of experimental configuration. The final data were obtained from ten independent experiments mean, table1 to table6 is as experimental results.

Table 1. Random Attacks Affect the Robustness of the Algorithm (MovieLens100K)

p^{fill}	3%		6%		9%		12%		15%	
p^{att}	MAE	PS	MAE	PS	MAE	PS	MAE	PS	MAE	PS
5%	0.73	0.01	0.72	0.01	0.01	0.01	0.73	0.01	0.72	0.01
10%	0.71	0.03	0.72	0.72	0.07	0.07	0.72	0.06	0.73	0.08
15%	0.72	0.11	0.72	0.17	0.18	0.18	0.18	0.17	0.72	0.16
20%	0.72	0.20	0.72	0.29	0.34	0.34	0.34	0.33	0.72	0.28

Table 2. Mean Attacks Affect the Robustness of the Algorithm (MovieLens100K)

p^{fill}	3%		6%		9%		12%		15%	
p^{att}	MAE	PS	MAE	PS	MAE	PS	MAE	PS	MAE	PS
5%	0.73	0.03	0.72	0.03.	0.72	0.73	0.72	0.04	0.73	0.05
10%	0.73	0.09	0.73	0.16	0.73	0.73	0.73	0.20	0.73	0.23
15%	0.73	0.33	0.72	0.39	0.73	0.74	0.74	0.51	0.75	0.53
20%	0.72	0.54	0.74	0.67	0.74	0.74	0.74	0.86	0.74	0.87

Table 3. Popular Attacks Affect the Robustness of the Algorithm (MovieLens100K)

p^{fill}	3%		6%		9%		12%		15%	
p^{att}	MAE	PS	MAE	PS	MAE	PS	MAE	PS	MAE	PS
5%	0.71	0.02	0.72	0.02	0.72	0.02	0.71	0.02	0.72	0.04
10%	0.71	0.14	0.73	0.13	0.72	0.12	0.72	0.11	0.73	0.10
15%	0.72	0.36	0.72	0.32	0.72	0.32	0.72	0.25	0.72	0.24
20%	0.71	0.63	0.72	0.50	0.72	0.50	0.72	0.45	0.72	0.44

Table 4. Random Attacks Affect the Robustness of the Algorithm (MovieLens1M)

p^{fill}	3%		6%		9%		12%		15%	
p^{att}	MAE	PS	MAE	PS	MAE	PS	MAE	PS	MAE	PS
5%	0.74	0.59	0.74	0.06	0.75	0.06	0.75	0.08	0.75	0.08
10%	0.74	0.98	0.75	0.13	0.75	0.16	0.75	0.15	0.75	0.17
15%	0.74	0.18	0.75	0.24	0.75	0.26	0.76	0.29	0.75	0.31
20%	0.75	0.28	0.76	0.33	0.76	0.37	0.76	0.42	0.75	0.40

Table 5. Mean Attacks Affect the Robustness of the Algorithm (MovieLens1M)

p^{fill}	3%		6%		9%		12%		15%	
p^{att}	MAE	PS	MAE	PS	MAE	PS	MAE	PS	MAE	PS
5%	0.74	0.07	0.74	0.09	0.73	0.13	0.74	0.17	0.74	0.17
10%	0.74	0.22	0.74	0.30	0.74	0.34	0.74	0.39	0.74	0.42
15%	0.74	0.39	0.73	0.53	0.73	0.60	0.73	0.60	0.74	0.64
20%	0.74	0.56	0.74	0.71	0.73	0.81	0.73	0.83	0.73	0.84

Table 6. Popular Attacks Affect the Robustness of the Algorithm (MovieLens1M)

p^{fill}	3%		6%		9%		12%		15%	
p^{att}	MAE	PS	MAE	PS	MAE	PS	MAE	PS	MAE	PS
5%	0.74	0.06	0.74	0.07	0.75	0.07	0.75	0.09	0.76	0.08
10%	0.74	0.15	0.75	0.19	0.75	0.19	0.75	0.20	0.75	0.20
15%	0.75	0.26	0.76	0.33	0.75	0.33	0.76	0.31	0.75	0.33
20%	0.75	0.38	0.75	0.43	0.76	0.44	0.76	0.46	0.76	0.43

Next, the fixed $p^{att} = 10\%$, taking p^{fill} as the variable, compared with CROFAP algorithm, contrast CROFAP algorithm, based on the user's recommendation algorithm, neighbor filtering algorithms and significant weighted average algorithm in robustness of the face of mean attack. Four belong to base on storage algorithm, with comparable, and with comparable. Figure 7 are referred to as CROFAP, KNN, NEIFIL and SIGWT. The design idea of filtering algorithm and CROFAP algorithm are similar, online mode to filter out attack from the nearest neighbor. The difference is that the latter uses adopt more fine-grained filtering. In single filtration, and allows other users to fill the vacancy left after filtering profile. Significant weighted algorithm to improve the calculation method of similar based on user recommendation algorithm. To stop the attack profile into the nearest neighbor effect, Figure 7 shows the experimental results.

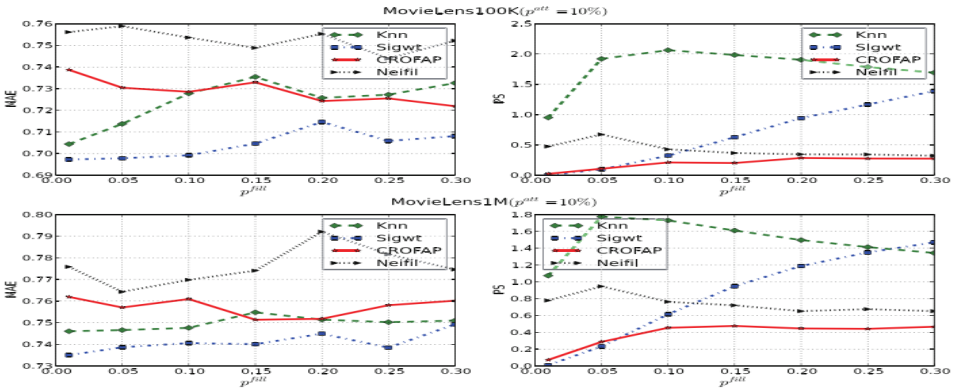

Figure 6. Comparison Robustness Algorithm of Mean Attack

5. Conclusion

This paper designed a kind of online filter of attack profile, using this method to improve nearest neighbor selection strategy based on user recommendation algorithm. Thus proposed

the collaborative recommendation algorithm based on attack profile online filter (CROFAP), Experiments show that in the face of random attack, mean attack and popular attack, online filter of attack profile can accurately identify and filter out the target user's nearest neighbor in most of the attack profile, to ensure effectively the robustness of the CROFAP algorithm.

References

[1] Z. Cheng and N. Hurley, "Effective Diverse and Obfuscated Attacks on Model-based-recommender Systems", In Proceedings of the 3rd ACM conference on Recommender systems. New York, USA, **(2009)**, pp. 141–148.
[2] N. Hurley, Z. Cheng and M. Zhang, "Statistical Attack Detection. In Proceedings of the 3rd ACM conference on Recommender systems", New York, New York, USA, **(2009)**, pp. 149–156.
[3] B. Mobasher, R. Burke, R. Bhaumik, *et al.*, "Toward Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness", ACM Transactions on Internet Technology, vol. 7, no. 4, **(2007)**.
[4] S. K. Lam and J. Riedl, "Shilling Recommender Systems for Fun and Profit", In Proceedings of the 13th international conference on World Wide Web. New York, NY, USA, **(2004),** pp. 393–402.
[5] J. L. Herlocker, J. A. Konstan, A. Borchers, *et al.*, "An Algorithmic Framework for Performing Collaborative Filtering", In Proceedings of the 22nd annual inter-national ACM SIGIR conference on Research and development in information retrieval. Berkeley, CA, USA, **(2009)** August, pp. 230–237.
[6] J. S. Breese, D. Heckerman and C. Kadie, "Empirical Analysis of Predictive Algorithm for Collaborative Filtering", In Proceedings of the 14th conference on Uncertainty in artificial intelligence, Madison, Wisconsin, USA, **(1998),** pp. 43–52.
[7] P.-A. Chirita, W. Nejdl and C. Zamfir, "Preventing Shilling Attacks in Online Recommender Systems", In Proceedings of the 7th ACM international workshopon Web information and data management, Bremen, Germany, **(2005)** November, pp. 67–74.

Author

Gao Feng, she is a lecturer at college of computer Changchun Normal University. She is in the research of computer network.

