International Journal of Future Generation Communication and Networking
Vol.7, No.2 (2014), pp.35-46
http://dx.doi.org/10.14257/ijfgcn.2014.7.2.04

A Collaborative Filtering Recommendation Algorithm Improved by Trustworthiness

Shengjun Xie

Campus network management center, Southwest University for Nationalities, Chengdu City, Sichuan, China
xieshj@swun.cn

Abstract

Recommender systems based on collaborative filtering have been well studied in both industry and academia fields. However, traditional collaborative filtering methods are typically in a low accuracy and lack of resistance towards attacks such as non-reliable information. To this end, in this paper, we propose a collaborative filtering recommendation algorithm improved by trustworthiness. Specifically, first we employ a content based method to identify a set of users with similar interests. Then, a trust model is applied as a scoring function, and higher ranked neighbors are selected as the evidence of prediction. Besides, we conducted extensive experiments using Netflix dataset, and the results show that our method is more efficient compared with others.

Keywords: *Collaborative filtering, Trust model, Prediction*

1. Introduction

As a popular technique for information filtering, recommender systems have been an efficient tool to deal with the problem of information overload all over the internet [1]. Existing recommendation techniques include content based, collaborative filtering, knowledge based and hybrid method. Recommender systems have been applied to many e-commerce enterprises, such as Amazon and eBay.

One of the most powerful technique is collaborative filtering (CF). The basic idea is: given a set of user-item ratings, the value for target user can be inferred from those who have similar interests with the target user. However, traditional collaborative filtering typically has two characteristics: (1) the rating matrix is very sparse, especially for large scale data oriented application; (2) and the prediction is made based upon nearest neighbors with similar interests. Based on above observations, traditional collaborative filtering algorithm typically produces a low accuracy and lacks of resistance towards noises such as non-reliable information.

To this end, in order to improve the accuracy of prediction based on CF and the resistance towards attacks of non-reliable information such as injected profiles, we propose a Collaborative Filtering recommendation algorithm Improved by Trustworthiness (CF-IT). Specifically, we perform two-step nearest neighbor selection, that is, neighbors are selected as those who (1) have similar interests with the target user, and (2) are trustworthy. In this way, our algorithm can achieve high accuracy and high resistance towards attacks and non-reliable information. The contributions of this paper can be summarized as follows:

(1) We proposed a neighbor selection strategy based on user similarity using topic models such as LDA [11];

(2) We developed a trust model to measure the trust relationship between target user and neighbor users with similar interests, in order to further identify the

ISSN: 2233-7857 IJFGCN

trustworthy neighbors for recommendation;

(3) Given a set of user-item rating information, based on above two-step nearest neighbor selection, we developed a CF-IF algorithm, which achieves to generate accurate and reliable predicted rating for target user and item pair;

(4) We conducted experiments on Netflix dataset, and compared our proposed algorithm with some existing methods. The results show that our method outperforms others with higher accuracy and resistance of attacks.

The remain of this paper is organized as follows. Section 2 provides some related works. Our proposed Collaborative Filtering recommendation algorithm Improved by Trustworthiness (CF-IT) method is discussed in Section 3. In Section 4, empirical experiments on Netflix dataset are conducted. In the end, Section 5 concludes this paper.

2. Related Work

Related efforts on collaborative filtering (CF) algorithms are roughly grouped into three categories: accuracy, robustness and trust oriented.

2.1. Accuracy Oriented CF Algorithms

Many researchers have proposed many modified algorithms to improve the accuracy of collaborative filtering. For instance, Chuang-Guang *et al.*, [2] presented a collaborative filtering algorithm based on uncertain neighbors. However, in real life application scenarios, it is very hard to compute factors for uncertain neighbors, which brings difficulties to balance the user set and item set, and therefore the accuracy of recommendation is impossible to be guaranteed. Cong *et al.*, [3] proposed a algorithm based on domain nearest neighbors. Their algorithm predicts the scores for users who are capable to recommend other users, which helps to improve the efficiency of selecting neighbors. However, if there exist non-reliable attacks such as injected profiles, the results would be greatly affected. Similarly, factorization based algorithm proposed by the Netflix prize winner [4] fails to provide the resistance of non-reliable information as well, even though the recommendation accuracy is improved.

2.2. Robustness Oriented CF Algorithms

Along the line of robustness, existing efforts are also made. For example, Mobasher *et al.*, [6] presented a model-based collaborative filtering as a defense against profile injection attacks. The algorithm is based on probabilistic latent semantic analysis (PLSA), and the recommendation is made after clustering over users or items. Sandvig *et al.*, [5] proposed to combine association rules mining and collaborative filtering, and achieved high robustness at the expense of relatively lower accuracy. Mehta *et al.*, [7] developed a singular value decomposition (SVD) based recommendation algorithm, which provided attack robustness by using a M-estimators function. However, this method works well in only small scale attacks.

2.3. Trust Oriented CF Algorithms

Besides, many researchers introduce trust into CF based recommendation, and proposed many trust models. For example, by studying on the trust relationship between users from a logistic perspective, Pitsilis *et al.*, [8] proposed a trust model based on uncertain probabilistic theory. However, since the uncertainty is calculated by average scoring for users and the scoring matrix is sparse, the accuracy of trust

model is limited. Kwon *et al.*, [9] developed a multi-dimensional trust model to analysis and measure expertise, trustworthiness and similarity, and then calculated a weighted summary to select neighbors. However, even though the proposed model is more diversity in choosing neighbors, it still lacks of resistance of attacks. Jamali *et al.*, [10] presented a trust model based on random walks to generate an aggregated score. But it is greatly affected by the sparsity of the scoring matrix.

To this end, to solve the above problems, in this paper, based on existing efforts, we propose a Collaborative Filtering recommendation algorithm Improved by Trustworthiness (CF-IT). Specifically, we perform two-step neighbors selection based on both similarity and trustworthiness, and then the target prediction is inferred from the optimal set of neighbors.

3. CF-IT Algorithm

In a CF system, suppose the set of users $U = \{u_1, u_2, \ldots u_m\}$, the set of items $I = \{i_1, i_2, \ldots, i_n\}$, and the user-item ratings is notated as matrix $\mathbf{R} : r_{i,j} (1 \le i \le m, 1 \le j \le n)$, where $r_{i,j}$ means the rating of user i on item j. The ratings for user i is $R(i) = \{r_{i1}, r_{i2}, \ldots, r_{in}\}$, and the ratings for item j is $R(j) = \{r_{1j}, r_{2j}, \ldots, r_{mj}\}$.

3.1. Choosing Similar Users

The recommendation is typically made by suggesting most nearest neighbors for target users. For example, in traditional KNN algorithm [12], if the selected K nearest neighbors have extremely low similarity with target users, the accuracy of recommendation would be poor. Therefore, choosing similar neighbors is significant. In this paper, we calculate the interest similarity between users and set a threshold as the filtering condition to identify similar candidates.

However, if the threshold is set to a fixed value, it is hard to be scalable and flexible for different applications. Therefore, in this paper, we dynamically set the threshold as the average of similarities between target user and all candidate neighbors. Suppose the set of all candidate neighbors as $N = \{u_1, u_2, \ldots, u_k\}$, and the similarity between target user u_t and $u_i (i = 1, 2, \ldots, k)$ is notated as $sim_{t,i}$. Therefore, the threshold setting for u_t is calculated as:

$$T_t = \frac{\sum_{i=1}^{k} sim_{t,i}}{k} \tag{1}$$

where k is the size of candidate neighbors.

Now the problem is how to calculate $sim_{a,i}$. A common practice is to use topic relevance [13, 14, 15]. Intuitively, content based relevance is captured by leveraging Vector Space Model (VSM) [16]. However, due to the large scale of our Netflix dataset, the dimension of word vectors is extremely high, which will increases the computing cost. Also, VSM does not distinguish the latent semantic meanings between words. Therefore, we propose to use topic models such as LDA [11] to calculate the interest similarity between users. The basic idea is to extract latent topics first and then build VSM vectors for similarity calculation.

Suppose the document is composed of user profile description and textual rating information over all items. Figure 1 illustrates the graphic representation of LDA,

where m is the size of users, D is the size of words within each document, and K is the number of latent topics. The objective of LDA model is to maximize the following function:

$$p(\mathbf{w} \mid \phi, \alpha, \beta) = \int \sum p(\mathbf{w} \mid z, \phi) p(z \mid \theta) p(\theta \mid \alpha) p(\phi \mid \beta) d\theta \tag{2}$$

where θ, ϕ are multinomial parameters over topics and words respectively, and α, β are predefined hyperparameters.

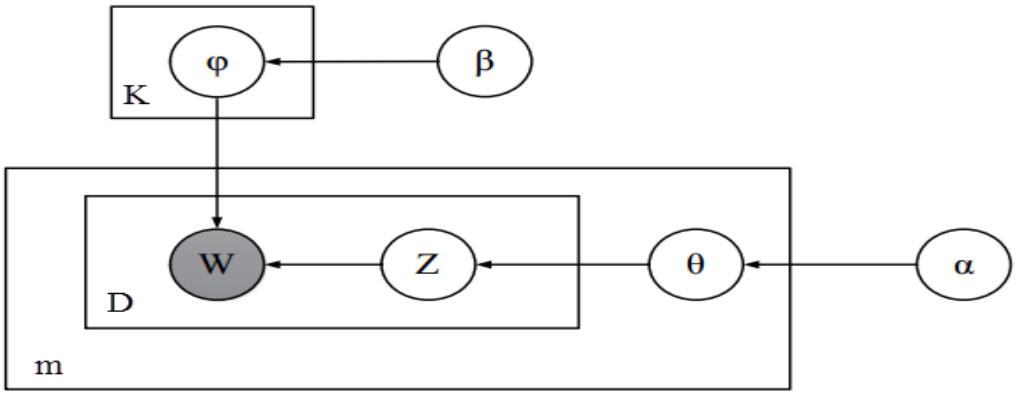

Figure 1. Graphic representation of LDA

After training process of LDA, we get the topic vector of user i:

$$\mathbf{T}_i = (p(z_1 \mid u_i), p(z_2 \mid u_i), \ldots, p(z_K \mid u_i)) \tag{3}$$

where z_i is the i-th topic, and $p(z_i \mid u_i) = \theta_{i,u_i}$ is learned from the model.

Next, we employ Kullback-Leibler (KL) divergence [17] to measure topic based interest similarity between users i, j:

$$sim_{i,j} = D_{KL}(\mathbf{T}_i \parallel \mathbf{T}_j) = \sum_{s=1}^{K} p(z_s \mid u_i) \log \frac{p(z_s \mid u_i)}{p(z_s \mid u_j)} \tag{4}$$

Suppose the target user is Tom, Table 1 illustrates the process of choosing similar users, where the value for user u_i over item i_j is the numeric ratings (or number of stars). The missing data means there is no rating for given user and item pair. Here we have all neighbors for Tom as $N = \{u_1, u_2, \ldots, u_6\}$. The threshold is calculated as (0.9431+0.4508-0.7826+0.5632+0.3335+0.5145)/6 = 0.3371. Therefore the selected neighbors based on interest similarity are $\{u_1, u_2, u_4, u_6\}$.

Table 1. Example of Choosing Similar Users

User	i_1	i_2	i_3	i_4	i_5	i_6	Similarity with Tom
Tom	5	1	2	3		3	
u_1	4	2	3	2	2	1	0.9431
u_2	3	4		3	3	2	0.4508

u_3	1	2	3	2	1	3	-0.7826
u_4	3	3	1		1	2	0.5632
u_5	4	3		2	3	1	0.3335
u_6	5		4		3	4	0.5145

3.2. Choosing Trustworthy Users

In this section, we discuss the second step, to choose trustworthy users based on similar neighbors. Since similar neighbors are selected based on content based information such as user profiles, our recommendation should also provides resistance of injected profile attacks.

In this paper we focus on the trustworthiness of users. Typically, there are two ways to calculate the trustworthiness of an entity: (1) based on historical interactions with target entity to predict future reliability, called *direct trustworthiness (DT)*; (2) inferred from the aggregated reliability from other entities, called *recommended trustworthiness (RT)*.

3.2.1. Direct Trustworthiness (DT): Suppose the trust of user i towards user j is $DT(i, j)$, calculated as:

$$DT(i, j) = \frac{1}{k}(\xi \sum_{s=1}^{k} S_s(i, j)t(s) + (1 - \xi) f(s) + p(s)) \quad (5)$$

Where k is the number of interactions between users i, j, and $S_s(i, j)$ means the degree of satisfactory at s-th interaction. $t(s)$ is a time attenuation function, $t(s) = \frac{1}{t_{now} - t_{interaction}}$. $f(s)$ is a context function for different application scenarios, values [0,1]. $p(s)$ is a penalty function, defined as:

$$p(s) = \theta(s)\frac{1}{1 + e^{-l}} \quad (6)$$

where l is the number of interaction failures, and

$$\theta(s) = \begin{cases} -1 & \text{failure on } s\text{-th interaction} \\ 0 & \text{succeed in } s\text{-th interaction} \end{cases} \quad (7)$$

3.2.2. Recommended Trustworthiness (RT): If there exist no interactions between user i and user j, recommended trustworthiness $RT(i, j)$ would be generated. For example, in Figure 2, the direction of arrows notate trust relationship, user i has direct trust relationships with $e_1, e_2, ..., e_n$, and $e_1, e_2, ..., e_n$ have direct direct trust relationships with user j. Therefore, $RT(i, j)$ could be calculated based on $e_1, e_2, ..., e_n$:

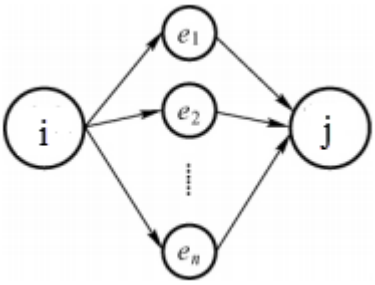

Figure 2. Recommended Trust Relationship

$$RT\ (i,\ j) = \frac{\sum_{s=1}^{|V|} DT\ (e_s,\ j)\omega_{ie_s}}{|V\ |} \tag{8}$$

where $|V\ |$ is the size of nodes who have interactions with j out of $\{e_1, e_2, ..., e_n\}$, $DT\ (e_s,\ j)$ means the direct trustworthiness from e_s to j, and ω_{ie_s} is the recommendation factor of e_s:

$$\omega_{ie_s} = \frac{S(i,\ j)}{N\ (i, e_s)} \tag{9}$$

Where $S(i,\ j)$ is the number of successful interactions between i and j, and $N\ (i, e_s)$ is the total number of interactions.

Therefore, trustworthiness between users $i,\ j$ is aggregated as $TW\ (i,\ j) = \alpha DT\ (i,\ j) + (1 - \alpha)RT\ (i,\ j)$, where α is a regulatory factor.

3.3. CF-IT Algorithm

After we have chosen similar and trustworthy neighbors, the next step is to infer rating for target user from selected neighbors.

Suppose the selected trustworthy similar neighbors for target user u_t from above two steps as $N\ (u_t)$, the predicted rating for u_t on item j is calculated as:

$$P_{t,\ j} = \bar{r}_t + \frac{\sum_{u_k \in N\ (u_t)} (r_{k,\ j} - \bar{r}_k) TW\ (t, k)}{\sum_{u_k \in N\ (u_t)} |\ TW\ (t, k)\ |} \tag{10}$$

where $r_{k,\ j}$ denotes the rating for user u_k on item j, $\bar{r}_t, \bar{r}_k$ denotes the average ratings for user u_t, u_k, and $TW\ (t,\ j)$ is the trust value of between target user u_t and neighbor u_k.

We summarize our proposed CF-IT algorithm, as shown in Figure 3.

Algorithm 1 CF-IT algorithm

Input: user-item rating matrix $\mathbf{R}$;
Output: rating for target user u_t on item j: $P_{t,j}$.

initialize trustworthy similar neighbors N_{u_t} as an empty set, and the size of all neighbors k
calculate similarity sim by Equation 4
set the threshold $T_t = \frac{\sum_{s=1}^{k} sim_{t,s}}{k}$
for $i = 1$ to k **do**
  if $sim_{t,i} > T_t$ **then**
    add u_i into N_{u_t}
    calculate $TW(t,i)$ by Equations 5 and 8
  end if
end for
sort N_{u_t} by $TW(t,i)$, and set N_{u_t} as the top 1 user
calculate $P_{t,j}$ by Equation 10
return $P_{t,j}$

Figure 3. Procedure of CF-IT Algorithm

4. Experiments

We use Netflix dataset for our experiments, which includes 480,189 users, 17,770 movies (i.e., items), and 103,297,638 ratings. We sampled 2,000 users, 4,000 items and 413,292 ratings in this experiment. We split the dataset into training set (90%) and test set (10%) randomly.

In order to measure the efficiency of our algorithm, we employ MAE (mean absolute error) as metric. The smaller MAE value is, the more accuracy the prediction is. MAE is calculated as follows:

$$MAE = \frac{\sum_{j=1}^{n} p_j - r_j}{n} \tag{11}$$

where p_j is predicted rating on item j, r_j is the ground truth rating, and n is the times of prediction.

We have three baseline algorithms for comparison. The first one is traditional Collaborative Filtering (CF), which makes the prediction based on KNN [12]. The second one is CF based on similar users by choosing neighbors with similar interests, as discussed in Section 3.1, denoted as CF-SU. The third one is CF based on user trustworthiness by choosing neighbors with most trustworthy neighbors, as discussed in Section 3.2, denoted as CF-UT. Our proposed algorithm is denoted as CF-IT.

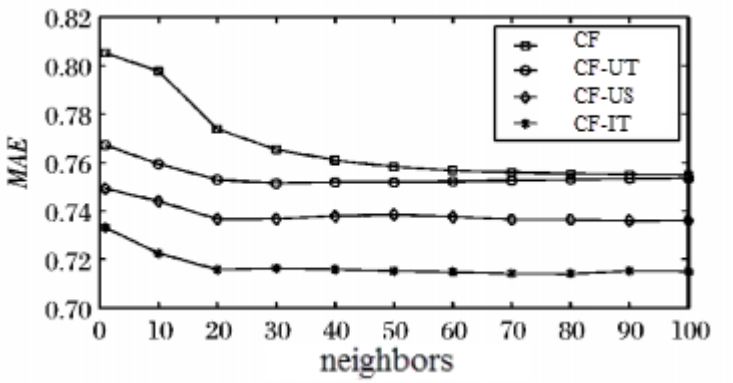

Figure 4. Comparison of MAE with Different Numbers of Neighbors

First, we investigate the accuracy of our proposed algorithm. Figure 4 shows the MAE values for four algorithms with different numbers of neighbors. We have two observations. First, as the number of neighbors grows, the accuracy of prediction decreases. Second, our CF-IT outperforms other algorithms. Specifically, tradition CF performs worst, and CF-IT works best. The results indicate that the combination of user similarity and trustworthiness is efficient.

Aside from accuracy, we also conduct experiments for the evaluation of attack resistance. The evaluation metrics are based on filler size and attack size. The former one means the set of items voted for attacks, and the latter means the number of injected profiles [18]. Both are randomly generated and measured in %, and the number of neighbors is set as 40. Table 2 lists the results for random attacks. Besides, we also depicts the prediction shift with 5% filler size in Figure 5. We have the following observations. First, the performance of CF based prediction is greatly affected by attacks, and the more attacks there are, the poorer the performance is. Second, compared to other algorithms, our proposed CF-IT exhibits best resistance towards attacks.

Table 2. Comparison of MAE with Different Sizes of Attacks

Attack size (%)	Filler size (%)	Algorithms		
		CF-UT	CF-US	CF-IT
5	1	0.9012	0.8924	0.8835
	3	0.8860	0.8686	0.8184
	5	0.8211	0.7938	0.7920
	10	0.7907	0.7759	0.7422
	20	0.7478	0.7336	0.7113
15	1	0.9144	0.9025	0.8989
	3	0.8864	0.8822	0.8705
	5	0.8793	0.8363	0.8538
	10	0.8347	0.8019	0.7983
	20	0.7998	0.7888	0.7508
20	1	0.9200	0.9111	0.9051

	3	0.9177	0.9065	0.8852
	5	0.8989	0.9012	0.8336
	10	0.8564	0.8500	0.8270
	20	0.8463	0.8402	0.8177
25	1	0.9509	0.9321	0.9209
	3	0.9226	0.9153	0.9187
	5	0.9150	0.8998	0.8940
	10	0.8911	0.8824	0.8745
	20	0.8741	0.8672	0.8287
30	1	0.9526	0.9433	0.9217
	3	0.9472	0.9304	0.9181
	5	0.9328	0.9148	0.9018
	10	0.9067	0.9039	0.8785
	20	0.8946	0.8711	0.8326
35	1	0.9877	0.9634	0.9367
	3	0.9731	0.9618	0.9215
	5	0.9691	0.9438	0.9166
	10	0.9487	0.9328	0.8641
	20	0.9107	0.9066	0.8416

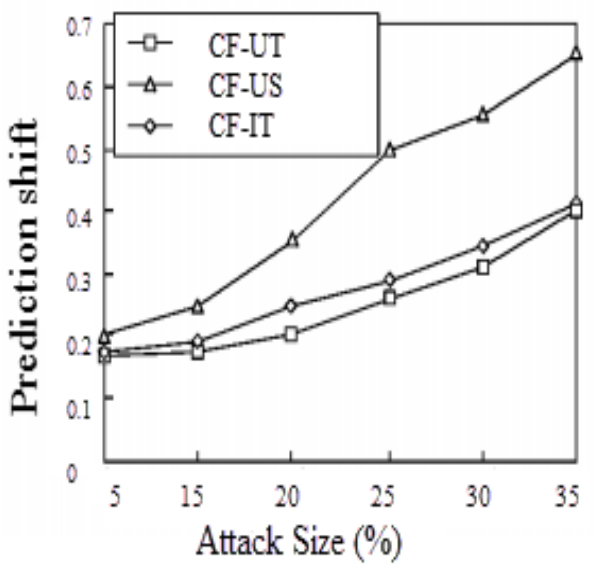

Figure 5. Results of Prediction Shift with 5% Filler Size

Last, in order to evaluate the performance of algorithm in different rating matrics with various sparsities, we randomly split the training and test sets as the

proportions of 4:6, 5:5, 6:4, and 7:3. Table 2 gives the results with various data sparsities. We can see that our CF-IT performs efficiently no matter how sparsity the dataset is.

Table 3. Comparison of MAE with Different Data Sparsities

Algorithms	4:6	5:5	6:4	7:3
CF	0.7788	0.7622	0.7614	0.7609
CF-UT	0.7599	0.7540	0.7426	0.7312
CF-US	0.7537	0.7433	0.7348	0.7228
CF-IT	0.7265	0.7198	0.7134	0.7025

5. Conclusion

Traditional collaborative filtering recommendation algorithms put more emphasize on the accuracy of recommendation and prediction, but fail to take into consideration of the trustworthy of entities. To his end, in this paper, we proposed a Collaborative Filtering recommendation algorithm Improved by Trustworthiness (CF-IT). Specifically, by adding a trust model to quantify the trustworthiness of users, we employ a two-step neighbors selection process, and accordingly nearest neighbors with similar interests and reliable as well are chose as the evidence to predict the rating for target user on given item: (1) the first step is to filter candidates with similar interests; and (2) the second step is to identify the most trustworthy neighbor within similar neighbors. Experiments conducted on Netflix dataset prove the efficiency and attack resistance, compared to some baseline methods.

However, in this paper, we assume the personal ability is identical for all users. In future work, we will try to investigate further the domain knowledge and experts influence on the neighbors selection process.

References

[1] H.-L. Xu, X. Wu, X.-D. Li and B. P. Yan, “Comparison study of Internet recommendation system”, Journal of software, vol. 20, no. 2, **(2009)**, pp. 350-362.
[2] H. U. A. N. G Chuang-Guang, Y. Jian, W. Jing, L. I. U. Yu-Bao and W. A. N. G. Jia-Hai, “Uncertain neighbors' collaborative filtering recommendation algorithm”, Chinese Journal of computers, vol. 33, no. 8, **(2010)**, pp. 1369-1377.
[3] C. Li, L. Changyong and M. Li, “A Collaborative Filtering Recommendation Algorithm Based on Domain Nearest Neighbor”, Journal of Computer Research and Development 9 (2008): 015.
[4] Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 426-434. ACM, 2008.
[5] J. J. Sandvig, B. Mobasher and R. Burke, “Robustness of collaborative recommendation based on association rule mining”, Proceedings of the 2007 ACM conference on Recommender systems, ACM, **(2007)**, pp. 105-112.
[6] B. Mobasher, R. Burke and J. J. Sandvig, “Model-based collaborative filtering as a defense against profile injection attacks”, AAAI, vol. 6, **(2006)**, pp. 1388.
[7] B. Mehta, T. Hofmann and W. Nejdl, “Robust collaborative filtering”, Proceedings of the 2007 ACM conference on Recommender systems, ACM, **(2007)**, pp. 49-56.
[8] Pitsilis, Georgios, and Lindsay Forsyth Marshall. A model of trust derivation from evidence for use in recommendation systems. University of Newcastle upon Tyne, Computing Science, 2004.
[9] K. Kwon, J. Cho and Y. Park, “Multidimensional credibility model for neighbor selection in collaborative recommendation”, Expert Systems with Applications, vol. 36, no. 3, **(2009)**, pp. 7114-7122.
[10] M. Jamali and M. Ester, “TrustWalker: a random walk model for combining trust-based and item-based recommendation”, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, **(2009)**, pp. 397-406.

[11] D. M. Blei, A. Y. Ng and M. I. Jordan, “Latent dirichlet allocation”, The Journal of machine Learning research, vol. 3, **(2003)**, pp. 993-1022.
[12] Jonathan L. Herlocker, J. A. Konstan, A. Borchers and J. Riedl, “An algorithmic framework for performing collaborative filtering”, Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, **(1999)**, pp. 230-237.
[13] M. Balabanović and Y. Shoham, “Fab: content-based, collaborative recommendation”, Communications of the ACM 40, no. 3, **(1997)**, pp. 66-72.
[14] Melville, Prem, Raymond J. Mooney, and Ramadass Nagarajan. "Content-boosted collaborative filtering for improved recommendations." In AAAI/IAAI, **(2002)**, pp. 187-192.
[15] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization”, In Proceedings of the fifth ACM conference on Digital libraries, ACM, **(2000)**, pp. 195-204.
[16] G. Salton, A. Wong and C.-S. Yang, “A vector space model for automatic indexing”, Communications of the ACM, vol. 18, no. 11, **(1975)**, pp. 613-620.
[17] S. Kullback, “Information theory and statistics”, Courier Dover Publications, **(1997)**.
[18] R. Burke, B. Mobasher, R. Zabicki and R. Bhaumik, “Identifying attack models for secure recommendation”, Beyond Personalization: A Workshop on the Next Generation of Recommender Systems, **(2005)**.

Author

Shengjun Xie, Male, Chengdu, sichuan. Master. Campus network management center, Southwest University for Nationalities. Engineer. Engaged in the study of network operation and management.

