Research Article

A Novel Preferential Diffusion Recommendation Algorithm Based on User's Nearest Neighbors

Fuguo Zhang,^{1,2} Yehuan Liu,¹ and Qinqiao Xiong¹

¹School of Information Technology, Jiangxi University of Finance & Economics, Nanchang 330013, China

²Research Institution for Information Resource Management, Jiangxi University of Finance & Economics, Nanchang 330013, China

Correspondence should be addressed to Fuguo Zhang; redbird_mail@163.com

Received 1 March 2017; Accepted 11 April 2017; Published 4 May 2017

Academic Editor: Hyo-Jong Lee

Copyright © 2017 Fuguo Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recommender system is a very efficient way to deal with the problem of information overload for online users. In recent years, network based recommendation algorithms have demonstrated much better performance than the standard collaborative filtering methods. However, most of network based algorithms do not give a high enough weight to the influence of the target user's nearest neighbors in the resource diffusion process, while a user or an object with high degree will obtain larger influence in the standard mass diffusion algorithm. In this paper, we propose a novel preferential diffusion recommendation algorithm considering the significance of the target user's nearest neighbors and evaluate it in the three real-world data sets: MovieLens 100k, MovieLens 1M, and Epinions. Experiments results demonstrate that the novel preferential diffusion recommendation algorithm based on user's nearest neighbors can significantly improve the recommendation accuracy and diversity.

1. Introduction

With the rapid development of Internet in the past years, the amount of online information increases at an exponential speed, which leads to information overload problem. When faced with vast amount of information, we can hardly find the valuable information accurately and quickly. The personalized recommender system is one of the most effective tools to resolve this problem, and it also can help enterprises make the users' potential demand a realistic demand [1, 2].

To date, various recommendation methods have been proposed and developed. One of the most successful recommender system methods is based on the collaborative filtering technique [3–5]. Recently, some physical methods, such as mass diffusion [6–9] and heat conduction [10, 11], have found applications in personalized recommendation. Standard mass diffusion algorithm applied the three-step mass diffusion starting from the target user on a user-object bipartite network, which accurately outperforms the standard collaborative filtering methods [1]. Many different bipartite network based methods [12] are proposed to achieve even better recommendation performance. In [6], Zhou et al. proposed

a hybrid method by combining the mass diffusion and heat conduction to solve the apparent diversity-accuracy dilemma of recommender systems. Motivated by enhancing the preferential diffusion algorithm's ability to find unpopular and niche objects, the preferential diffusion has been designed in [9]. Moreover, Zhang and Zeng proposed a strategy to adding some virtual connections to the networks, which is useful to deal with the cold start problem in recommender system [13].

However, all these methods do not give a high enough weight to the influence of the target user's nearest neighbors in the resource diffusion process. As we all know, birds of a feather flock together. The user's nearest neighbors are the ones who have similar taste with the given user. Therefore we introduce a novel preferential diffusion recommendation algorithm considering the significance of the target user's nearest neighbors in the diffusion process.

2. Methods

A recommender system can be represented by a bipartite network $G(U, O, E)$, where $U = \{u_1, u_2, \dots, u_m\}$, $O = \{o_1, o_2, \dots, o_n\}$, and $E = \{e_1, e_2, \dots, e_q\}$ are the sets of users, objects,

FIGURE 1: Standard mass diffusion algorithm at work on the bipartite user-object network. Users are shown as circles and objects are squares. The target user is indicated by the black circle.

and links, respectively [7]. Denote by $A_{m \times n}$ the adjacency matrix, where the element $a_{ia} = 1$ if the user i has selected the object a and $a_{ia} = 0$ otherwise.

2.1. Standard Mass Diffusion Recommendation Algorithm. As is shown in Figure 1, the standard mass diffusion (SMD) algorithm is equivalent to a three-step random walk process. At first, objects in the bipartite network are assigned an initial resource f , with $f^i = \{f_1^i, f_2^i, \dots, f_\alpha^i, \dots, f_n^i\}$ for the target user i . For simplicity, if an object is collected by the user i , its initial resource is assigned to be 1, otherwise it is assigned to be 0. That is to say, the initial resource vector f can be written as

$$f_\alpha^i = a_{i\alpha}. \quad (1)$$

Then, each object's resource is redistributed to the user who has collected the object averagely, and the user's resource is the sum of the resources received from objects. At last, each user's resource was reallocated to the objects which he has collected averagely. The final score of the object's resource can be calculated via the transformation $f' = Wf$, where W is the resource transfer matrix.

$$w_{\alpha\beta} = \frac{1}{k_\beta} \sum_{l=1}^m \frac{a_{l\alpha} a_{l\beta}}{k_l}, \quad (2)$$

where k_β is the degree of the object β and k_l is the degree of the user l .

2.2. The Novel Preferential Diffusion Algorithm Based on User's Nearest Neighbors. Following on from previous research [14], the diffusion process of the novel preferential diffusion recommendation algorithm based on user's nearest neighbors (NNMD) is shown in Figure 2. At first, we calculate the Jaccard similarities between the target user i and the other users to get the top N similar neighbors. The formula of Jaccard similarity reads

$$J_{ij} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}, \quad (3)$$

where J_{ij} is the Jaccard similarity between user i and user j and N_i and N_j are the user neighbors set of user i and user j , respectively. Then we can get the objects' initial resource

denoted by the vector f_1 , with $f_1^i = \{f_{11}^i, f_{12}^i, \dots, f_{1\alpha}^i, \dots, f_{1n}^i\}$ for the target user i . f_1 can be written as

$$f_{1\alpha}^i = a_{i\alpha} + \sum_{k \in U} a_{k\alpha}, \quad (4)$$

where U is the nearest neighbors set of the target user i . In Figure 2, $f_1 = (3, 1, 2, 2, 0)$. But only the objects which the target user i has selected can distribute the resources to users and then redistribute them via the transformation

$$f_2 = W(f_1 f), \quad (5)$$

where W is the same as (2). In Figure 2, $f_2 = (2, 0.5, 1, 1.5, 0)$. Finally, we use the linear combination the resources vectors f_1 and f_2 to get the last objects' resources vector F . That is to say,

$$F = \alpha f_1 + (1 - \alpha) f_2, \quad (6)$$

where α is a variable parameter from 0 to 1.

3. Data and Metrics

3.1. Data. To test the algorithmic performance, we use three benchmark data sets as shown in Table 1. The sparsity of these data sets is shown in the last column of Table 1. They are very sparse, especially Epinions data set. MovieLens 100k and MovieLens 1M data sets [15] were collected by the GroupLens research group. They consist of 100000 ratings from 943 users on 1682 different movies and 1000209 ratings from 6040 users on 3952 different movies, respectively. The ratings are integer numbers in the range of 1 to 5 scales. The Epinions data set [16] consists of 22166 users, 296277 objects, and 922267 ratings. It is noted that Epinions data set is highly sparse. Users only rate a small number of items in the system, and, in order to get better results, we delete those users and objects with degree less than 7. Finally, we get a new data set which consists of 4066 users, 7649 objects, and 154122 ratings. We randomly divide the data sets into two parts: the training set E^T contains 80% of the data and the remaining 20% of data constitutes the probe set E^P .

3.2. Metrics. There has been considerable research in the area of recommender systems evaluation. Accuracy is the most important aspect in evaluating the recommendation algorithmic performance. In this paper, we use ranking score [8] to measure the ability of a recommendation algorithm to generate a ranking list of the target user's uncollected objects that matches the users' preference. For the target user u_i , the recommendation algorithm will return u_i a ranking list of all his unselected objects and, according to E^P , if u_i has selected the object o_j and o_j is at r_{ij} th place in the ranking list, we say the position of o_j is

$$R_{ij} = \frac{r_{ij}}{L}, \quad (7)$$

FIGURE 2: A novel preferential diffusion algorithm by user's nearest neighbors at work on the bipartite user-object network. Users are shown as circles and objects are squares. The target user is indicated by the black circle and the nearest neighbors of the target user are the circles which have a letter "n" in them and α is the variable parameter.

TABLE 1: Basic properties of the three data sets and the sparsity is defined as $E/(N_u N_o)$.

Network	E	N_u	N_o	Sparsity
MovieLens 100k	100000	943	1682	0.063
MovieLens 1M	1000209	6040	3952	0.042
Epinions	154122	4066	7649	0.005

TABLE 2: The four cases of the unselected objects of the target user in the recommender system.

User likes	Recommender system recommended	Recommender system did not recommend
Likes	C_{tp}	C_{fn}
Does not like	C_{fp}	C_{tn}

where L is the number of his unselected objects. We obtain the mean value of all the user-object ranking scores in E^P ; namely,

$$R = \frac{1}{|E^P|} \sum_{ij \in E^P} R_{ij}. \quad (8)$$

Clearly, the larger the ranking score, the lower the algorithm's accuracy and vice versa.

In the practical recommender system, we may consider the number of objects that users like in the recommendation list. Therefore, we take another accuracy metric called precision. For a target object o_j and user u_i , there are four cases in the recommender system. The first is that the recommender system recommended the object and user likes it. The second is that recommender system recommended the object but the user does not like it. The third is that the user likes the object but the recommender system did not recommend it. Finally is the case that the user does not like the object and the recommender system did not recommend it. As is shown in Table 2, C_{tp} , C_{fn} , C_{fp} , and C_{tn} denote the number of the objects in the four cases.

For a target user u_i , the precision of recommendation $P_i(L)$ is defined as

$$P_i(L) = \frac{C_{tp}}{L} = \frac{C_{tp}}{C_{tp} + C_{fp}}. \quad (9)$$

We obtain the mean precision $P(L)$ of all the users in the recommender system. Besides accuracy, diversity is taken into account as another important aspect to evaluate the recommendation algorithm. There are two kinds of diversity. One is called intrauser-diversity [17]; the other is called interuser-diversity [18]. In this paper, we consider the interuser-diversity. It considers the different objects between users in the recommendation list. For two users u_i and u_j , the differences can use be measured by the Hamming distance [18]:

$$H_{ij}(L) = 1 - \frac{S_{ij}(L)}{L}, \quad (10)$$

where $S_{ij}(L)$ is the number of common objects between u_i and u_j in the recommendation list and L is the length of the recommendation list. Clearly, if u_i and u_j have the same recommendation list, $H_{ij}(L) = 0$, while if the recommendation lists are completely different, $H_{ij}(L) = 1$.

In reality, it has been found that a recommender system which has a high accuracy might not be satisfied by the users [19]. For example, for a film website, recommending the popular films to the users may not always be the best recommendation, because users might have already seen those films in other ways. A good recommender system can find the objects that match the users' preferences and are unlikely to be already known. As a result, the novelty is also often used in evaluating the recommendation algorithmic performance.

The average degree of objects in the recommendation list is widely used to identify the novelty of a recommender system [20], which is defined by

$$N(L) = \frac{1}{ML} \sum_u \sum_{o_j \in O_R^i} k_{o_j}, \quad (11)$$

where M is the number of users, O_R^i is the recommendation list for user u_i , and k_{o_j} is the degree of the object o_j .

TABLE 3: Algorithmic performance for MovieLens 100k, MovieLens 1M, and Epinions data. The precision, interuser-diversity, and novelty are corresponding to $L = 20$. The parameters for NNMD are $N = 50$ and $\alpha = 0.9$ in MovieLens 100k and MovieLens 1M, while, in Epinions, the parameters are $N = 10$ and $\alpha = 0$. The entries corresponding to the best performance over all methods are emphasized in bold.

Data set	Algorithms	Ranking score	Precision	Interuser-diversity	Novelty
MovieLens 100k	NNMD	0.059537	0.2242	0.8401	237
	SMD	0.069011	0.1971	0.6970	279
MovieLens 1M	NNMD	0.077039	0.2726	0.8816	1340
	SMD	0.095269	0.1949	0.5865	1828
Epinions	NNMD	0.180439	0.0374	0.6787	204
	SMD	0.181141	0.0357	0.6743	205

FIGURE 3: The overall ranking score of the NNMD algorithm under different N and α in MovieLens 100k, MovieLens 1M, and Epinions data set and the ranking score of the SMD algorithm in MovieLens 100k, MovieLens 1M, and Epinions.

4. Results and Discussion

In our first set of experiments, we compare the ranking score of the NNMD algorithm under different α and top N (N is the number of the target user's nearest neighbors) with that of the SMD algorithm. The results on MovieLens 100k, MovieLens 1M, and Epinions data are reported in Figure 3. Clearly, we can see that in MovieLens 100k and MovieLens 1M, with the increase of N , the rank score is smaller and smaller; that is to say, the recommendation accuracy is getting better and better. However, when N is more than 30, the change of rank score is very small. Moreover, as long as α is not equal to 0 or 1, the rank score of our method is better than that of the SMD algorithm. It is interesting to note that the optimal parameters of our method are the same in MovieLens 100k and MovieLens 1M, which are $N = 50$ and $\alpha = 0.9$, while, in Epinions, the improvement of the rank score is not significant. When α is greater than 0 or N is greater than 20 the rank score of the NNMD algorithm is a little worse than

that of the SMD algorithm, and, with the change of α and N , the rank scores of the two algorithms are almost the same. But when N is less than 20 and $\alpha = 0$, the rank score of our method is getting better than that of the SMD algorithm. Clearly, we can get the optimal parameters $N = 10$ and $\alpha = 0$ in Epinions.

Then we examined the performance in precision, interuser-diversity, and novelty of our novel algorithm at the optimal parameters N and α . Summaries of the results for all algorithms and metrics on MovieLens 100k, MovieLens 1M, and Epinions data sets are shown in Table 3. The optimal parameters are subject to the lowest ranking score. The other three metrics, namely, precision, interuser-diversity, and novelty, are obtained at the optimal parameters. Clearly, the NNMD algorithm outperforms the SMD algorithm over all four evaluation metrics.

The comparison of precision between NNMD and SMD in three data sets under different length of recommendation list is shown in Figure 4. It clearly indicates that the precision

FIGURE 4: The precision of NNMD and SMD algorithm in MovieLens 100k, MovieLens 1M, and Epinions under different length of recommendation list. The parameters for the NNMD algorithm are $N = 50$ and $\alpha = 0.9$ in MovieLens 100k and MovieLens 1M while they are $N = 10$ and $\alpha = 0$ in Epinions.

FIGURE 5: The interuser-diversity of the NNMD and SMD algorithm in MovieLens 100k, MovieLens 1M, and Epinions under different length of recommendation list. The parameters for the NNMD algorithm are $N = 50$ and $\alpha = 0.9$ in MovieLens 100k and MovieLens 1M while they are $N = 10$ and $\alpha = 0$ in Epinions.

of the NNMD algorithm is better than that of the NMD algorithm in all the three data sets and it has a very significant improvement in MovieLens 100k and MovieLens 1M. That is to say, our method can recommend objects for users more accurately.

Figure 5 shows the comparison of interuser-diversity between our method NNMD and SMD in three data sets under different length of recommendation list. It clearly shows that interuser-diversity of our NNMD algorithm is better than that of the SMD algorithm in all the three data sets, especially in MovieLens 100k and MovieLens 1M. In

other words, the objects in the recommendation list of our method are more different between users.

Figure 6 shows the comparison of novelty between our method NNMD and SMD in three data sets under different length of recommendation list. It clearly indicates that the novelty of our method is much better than the SMD in MovieLens 100k and MovieLens 1M, while, in Epinions, the results of the two algorithms are very similar, but our method also has a little improvement than that of the SMD algorithm.

In summary, the recommendation performance of our method is better than that of the standard mass diffusion. In

FIGURE 6: The novelty of NNMD and SMD algorithm in MovieLens 100k, MovieLens 1M, and Epinions under different length of recommendation list. The parameters for the NNMD algorithm are $N = 50$ and $\alpha = 0.9$ in MovieLens 100k and MovieLens 1M, while they are $N = 10$ and $\alpha = 0$ in Epinions.

particular, the precision of our method increases an average of 13.27% percent compared to that of the SMD in MovieLens 100k and increases an average of 35.9% percent in MovieLens 1M and increases an average of 4.47% percent in Epinions. Although the improvement of the algorithmic performance in some aspects is not significant in Epinions data set, the reason may be that the data is so sparse that the novel algorithm cannot get the proper user's nearest neighbors and it affects our algorithmic performance.

5. Conclusion and Future Work

Most of network based recommendation algorithms have a tendency to recommend popular objects to the users [1] because the object with high degree has a significant influence in the resource diffusion process. In this paper we propose a novel preferential diffusion recommendation algorithm based on user's nearest neighbors which give a high weight to the influence of the target user's nearest neighbors in the resource diffusion process. Experimental results based on MovieLens 100k, MovieLens 1M, and Epinions data set show that making a suitable adjustment in the parameter α or the size of the user's nearest neighbors set can help recommendation algorithm get a better recommendation performance. It can not only provide more accurate recommendations but also generate more diverse and novel recommendations.

For future work, we intend to consider the level of rating between user and his nearest neighbors. Moreover, we will use the trust data [21, 22] in the network, because it can be used to find the nearest neighbors more accurately in high sparse data set, and it may have a better recommendation performance.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (Grant nos. 71361012 and 71363022), by National Science Foundation of Jiangxi, China (no. 20161BAB201029), and by the Foundation of Jiangxi Provincial Department of Education (no. GJJ. 150446).

References

- [1] T. Zhoua, Z. Kuscsik, J. Liu, M. Medo, J. R. Wakeling, and Y. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 10, pp. 4511–4515, 2010.
- [2] L. Lü, M. Medo, C. H. Yeung, Y. Zhang, Z. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [3] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.
- [4] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998.
- [5] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [6] T. Zhou, J. Ren, M. Medo, and Y. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, Article ID e046115, 2007.
- [7] M. S. Shang, L. Lu, Y. C. Zhang, and T. Zhou, "Empirical analysis of web-based user-object bipartite networks," *Europhysics Letters*, vol. 90, no. 4, Article ID e48006, 2010.

- [8] A. Zeng, A. Vidmer, M. Medo, and Y.-C. Zhang, "Information filtering by similarity-preferential diffusion processes," *EPL*, vol. 105, no. 5, Article ID e58002, 2014.
- [9] L. Lu and W. Liu, "Information filtering via preferential diffusion," *Physical Review E*, vol. 83, no. 6, Article ID e066119, 2011.
- [10] Y. C. Zhang, M. Blattner, and Y. K. Yu, "Heat conduction process On community networks as a recommendation model," *Physical Review Letters*, vol. 99, no. 10, Article ID 154301, 2007.
- [11] J. G. Liu, T. Zhou, and Q. Guo, "Information filtering via biased heat conduction," *Physical Review E*, vol. 84, no. 3, Article ID e037101, 2011.
- [12] F. Yu, A. Zeng, S. Gillard, and M. Medo, "Network-based recommendation algorithms: a review," *Physica A: Statistical Mechanics and Its Applications*, vol. 452, pp. 192–208, 2016.
- [13] F. Zhang and A. Zeng, "Improving information filtering via network manipulation," *Europhysics Letters*, vol. 100, no. 5, Article ID 58005, 2012.
- [14] F. G. Zhang, Y. H. Liu, and Q. Q. Xiong, "A novel mass diffusion recommendation algorithm based on user's nearest neighbors," in *Proceedings of the International Symposium on Information Technology Convergence*, 2016.
- [15] <http://www.grouplens.org/>.
- [16] <http://www.epinions.com/>.
- [17] T. Zhou, R. Q. Su, R. R. Liu, L. L. Jiang, B. H. Wang, and Y. Zhang, "Accurate and diverse recommendations via eliminating redundant correlations," *New Journal of Physics*, vol. 11, Article ID 123008, 2009.
- [18] T. Zhou, L. L. Jiang, R. Q. Su, and Y. C. Zhang, "Effect of initial configuration on network-based recommendation," *Europhysics Letters*, vol. 81, no. 5, pp. 58004–58007, 2008.
- [19] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: evaluating recommender systems by coverage and serendipity," in *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*, pp. 257–260, Barcelona, Spain, September 2010.
- [20] Z.-K. Zhang, C. Liu, Y.-C. Zhang, and T. Zhou, "Solving the cold-start problem in recommender systems with social tags," *Europhysics Letters*, vol. 92, no. 2, Article ID 28002, 2010.
- [21] C. Martinez-Cruz, C. Porcel, J. Bernabé-Moreno, and E. Herrera-Viedma, "A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling," *Information Sciences*, vol. 311, pp. 102–118, 2015.
- [22] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1763–1777, 2014.