

Research Article

Personalized Recommendation via Suppressing Excessive Diffusion

Guilin Chen, Xuzhen Zhu, Zhao Yang, and Hui Tian

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Hui Tian; tianhui@bupt.edu.cn

Received 15 November 2016; Revised 1 March 2017; Accepted 23 May 2017; Published 18 June 2017

Academic Editor: Yann Favennec

Copyright © 2017 Guilin Chen 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.

Efficient recommendation algorithms are fundamental to solve the problem of information overload in modern society. In physical dynamics, mass diffusion is a powerful tool to alleviate the long-standing problems of recommendation systems. However, popularity bias and redundant similarity have not been adequately studied in the literature, which are essentially caused by excessive diffusion and will lead to similarity estimation deviation and recommendation performance degradation. In this paper, we penalize the popular objects by appropriately dividing the popularity of objects and then leverage the second-order similarity to suppress excessive diffusion. Evaluation on three real benchmark datasets (MovieLens, Amazon, and RYM) by 10-fold cross-validation demonstrates that our method outperforms the mainstream baselines in accuracy, diversity, and novelty.

1. Introduction

With the rapid development of Internet technology and the explosion of information, information overload is increasingly exacerbated and can not be ignored [1]. That massive data makes it difficult for people to obtain the most relevant information promptly has turned into a hindrance to the development of Internet technology [2]. Personalized recommendation [3–5] is fundamental to solve the problem of information overload which adopts previous interactions records to extract users' interests for making recommendations. Despite the fact that vast amount of work has been done [5, 6], it is still far from enough to satisfy the increasing needs of commodity information service [7–10].

Due to the urgent demands of the E-economy, various recommendation algorithms have been proposed: content-based (CB) approach [10] captures user's preferences to recommend similar objects, but it does not apply to audio, image, or video information; spectral analysis (SA) [4] is not suitable for huge-size systems because of high computational complexity; collaborative filtering (CF) [11, 12] is based on similarity and consequently suffers from a popularity bias problem; network-based (NB) approach [13–15] constructs a network based on the relationships between the users and

the objects and then analyzes the network to recommend for users but there exists the “cold-start” problem. In this paper, motivated by mass diffusion [13, 16–18] we focus on recommendation methods that directly build on a network representation of the input data. This elementary approach has been modified and generalized many times since then in order to improve the accuracy and diversity of the recommendations [19].

In this paper, the problem widely existing in mass diffusion-based recommendation algorithms is raised as “excessive diffusion” (defined in Section 2) and we propose a novel algorithm to relieve it. Popularity bias [19] and redundant similarity [20] can both be attributed to excessive diffusion which will definitely lead to overestimated similarity and depress recommendation performance. With the motivation of improving the recommendation performance, in this paper, we firstly penalize the weight of popular objects and consider the heterogeneity of users' degrees simultaneously to restrain popularity bias and then we leverage the second-order similarity to further eliminate redundant similarities. Extensive experiments on three real datasets (MovieLens, Amazon, and RYM) demonstrate the effectiveness of the proposed algorithm in improving recommendation accuracy, diversity, and novelty thus applicable in practice.

2. Related Work

Theoretical physics has provided us with some useful tools to improve the recommendation performance, such as mass diffusion (MD) [16–18] and heat conduction (HC) [21, 22]. Network-Based Inference (NBI) [13] is a classical recommendation algorithm based on mimicking the mass diffusion resource-allocation process between objects via neighboring users, which is biased towards popular objects. There are a lot of variants of it: Heterogenous Network-Based Inference (HNBI) [16] initiates the resource distribution heterogeneously; Redundant Eliminated Network-Based Inference (RENBI) [23] eliminates redundant correlations by considering high order correlations between objects; Corrected Similarity-based Inference (CSI) [24] combines the forward and backward diffusion to correct similarity estimation deviation and so forth. HC [21] imitates the process of heat conduction, which achieves high diversity but low accuracy. Combining the MD and HC processes is an intuitive idea and there are many hybrid algorithms that have been developed to decrease the NBI's bias towards popular objects, like hybrid heat-spreading and probabilistic-spreading (HHP) [25], balanced diffusion (BD) [26], preferential diffusion (PD) [27], similarity-preferential hybrid processes (SPHY) [28], and so on [19, 23].

2.1. The Problem of Excessive Diffusion. The essential point that hinders the improvement of recommendation performances lies in the bias on some objects. Putting it in the diffusion paradigm, excessive resources are spread to them. That is why we call it excessive diffusion. Although there have been many successful improvements, excessive diffusion is ubiquitous in realistic recommendation systems. The appearance it takes on is mainly manifested in two aspects: on the one hand, the resources diffusion is excessive biased towards popular objects making the popular objects more likely to be recommended, which leads to popularity bias and will not necessarily promote the accuracy but undermine recommendation diversity and novelty. On the other hand, the resources derived from the same user may be repeatedly counted in the diffusion process; that is, excessive resources are distributed to the object, which results in redundant similarity. An illustration of redundant similarity is shown in Figure 1. Excessive diffusion causes some objects to have too many resources than they deserve which seriously depresses the recommendation performance and should be suppressed to improve the recommendation performance.

3. Method

Suppose that a recommendation system contains m users and n objects and each user has collected some objects. Let $U = \{u_1, u_2, \dots, u_m\}$ and $O = \{o_1, o_2, \dots, o_n\}$ represent the users and objects, respectively. According to user's purchase history we can construct the user-object bipartite network with $m+n$ nodes. If o_j is collected by u_i , there is a link between u_i and o_j , and the corresponding element a_{ij} in the adjacent matrix A is set as 1, otherwise 0. Mathematically speaking, the essential task of a recommendation system is to generate a ranking list

of the target user's uncollected objects. The top L objects are recommended to this user.

Network-Based Inference (NBI) [13] is based on simulating the mass diffusion resource-allocation process, where each object distributes its initial resource equally to all the users who have collected it, and then each user equally reallocates what he/she has received to all the objects he/she has collected, the transfer weight w_{ij} can be defined as

$$w_{ij} = \frac{1}{k(o_j)} \sum_{l=1}^m \frac{a_{li} a_{lj}}{k(u_l)}, \quad (1)$$

where $k(u_l)$ represents the number of objects collected by u_l and $k(o_j)$ denotes the number of users who have collected o_j .

For a specific user u_i , we initially assign each object that has been collected by u_i one unit of resource, while others are assigned 0; namely, $f_j = a_{ij}$, and then we redistribute these resources via the transformation $f' = Wf$, where W is the transfer matrix, $f = (f_j)_{n \times 1}$, and f' represents the final resource distribution of objects and the top- L values of uncollected objects will be recommended to u_i .

Motivated by enhancing the algorithm's recommendation performance, we find a suitable way to suppress excessive diffusion. We penalize the popular objects by assigning more resource to the low-degree objects at the last step (a user redistributes his/her resource to his/her neighbor object o_i the amount proportional to $k(o_i)^\beta$) and consider the heterogeneity of users' degrees simultaneously [27] to mitigate popularity bias and strengthen the capability of finding unpopular and niche objects. In this case, (1) is transformed into

$$\begin{aligned} w_{ij} &= \frac{1}{k(o_j) k(o_i)^{-\beta}} \sum_{l=1}^m \frac{a_{li} a_{lj}}{k(u_l) E(a_{lr} k(o_r)^\beta)} \\ &= \frac{1}{k(o_j) k(o_i)^{-\beta}} \sum_{l=1}^m \sum_{r=1}^n \frac{a_{li} a_{lj}}{a_{lr} k(o_r)^\beta}, \end{aligned} \quad (2)$$

where $-1 \leq \beta \leq 0$ is a free parameter and $E(a_{lr} k(o_r)^\beta)$ denotes the average of $k(o_r)^\beta$ over all objects that have been collected by user u_l .

As we know, some resources that object obtained may be derived from diverse users and others may stem from the same user. Considering the example in Figure 1, from our point of view, A and B are independent of C in the sense that they appear in distinct transactions that have C (no common user collecting A, B, and C together) and therefore their adjacent relations in the graph with C are independent. Then, the second-order similarity between A and C, derived from the path $A \rightarrow B \rightarrow C$, should be insignificant because A and B are independent of C. On the contrary, D and F have strong second-order similarity in that they appear in the same transactions that have F (D, E, and F are collected by the same user), which leads to strong correlation between both. That is, if there exists redundant similarity between two specific objects, the second-order similarity of them should be strong. Hence, we can subtract the second-order similarities in an

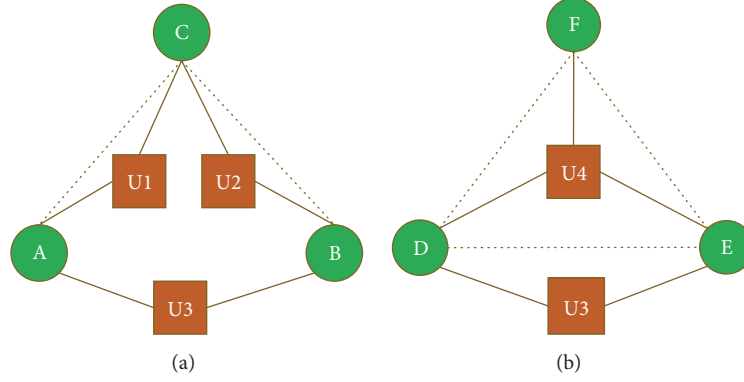


FIGURE 1: Illustration of redundant similarities. Objects and users are marked with circles and squares, respectively. The solid line represents the object that has been collected by user and the dotted line denotes the similarity between objects. As we can see that the resources of C obtained through the diffusion following paths $A \rightarrow \text{user } 1 \rightarrow C$ and $B \rightarrow \text{user } 2 \rightarrow C$ are independent for there is no common user collecting A, B, and C together in Figure 1(a), however, in Figure 1(b), the resources of F are obtained from the same user and are counted twice. If user 4 chooses F just because F has some similarities with D (or E), the similarity E-F (or D-F) is redundant when calculating the resource of F, which degrades the recommendation accuracy.

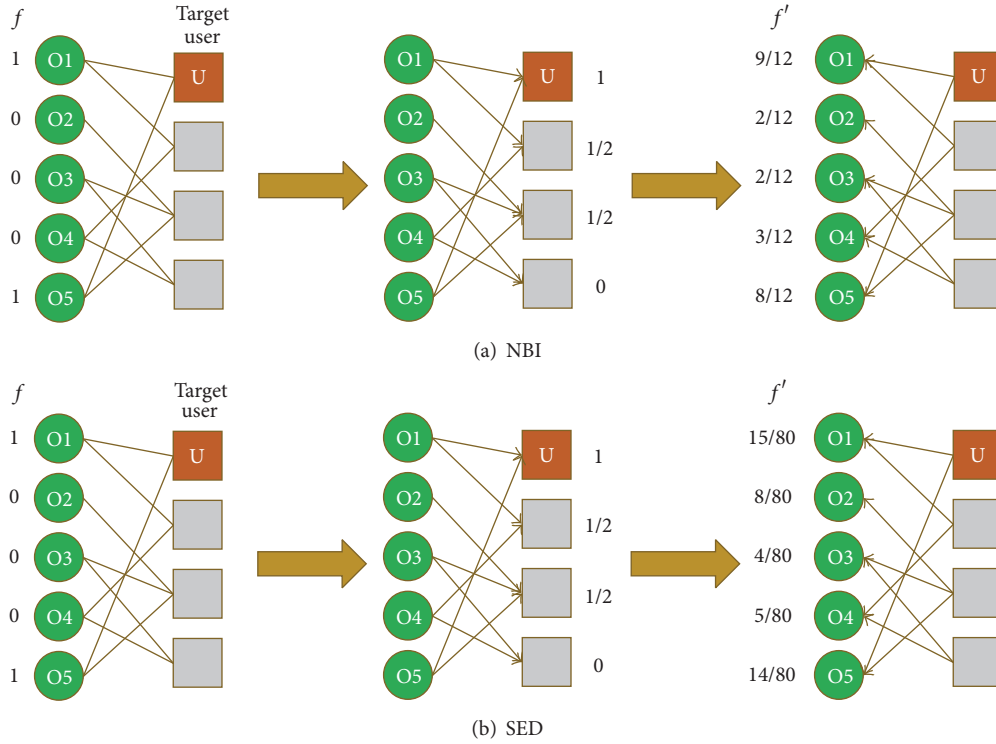


FIGURE 2: Illustrations of NBI and SED on user-object bipartite networks. Objects and users are represented by circles and squares, respectively. In SED algorithm, we set $\alpha = 0$ and $\beta = -1$.

appropriate way to eliminate redundant similarities to some extent [20]. As a result, we design an improved algorithm via suppressing excessive diffusion (SED), and the transfer matrix is modified as

$$W^{\text{SED}} = W + \alpha W^2, \quad (3)$$

where $\alpha \leq 0$ is a free parameter and $W = (w_{ij})_{n \times n}$ is achieved via (2). Clearly, when $\alpha = 0$ and $\beta = 0$, it will degenerate

to NBI. Figure 2 gives illustrations about how to calculate transfer matrix of NBI and SED.

4. Data and Evaluation

4.1. Dataset. Three real datasets, including MovieLens (<http://www.grouplens.org/>), Amazon (<http://www.amazon.com/>), and RYM (<http://rateyourmusic.com/>), are employed to test the algorithm's performance. They all leverage ratings

TABLE 1: Primary information of the three datasets.

Data	Users	Objects	Links	Sparsity
MovieLens	943	1682	82520	6.3×10^{-1}
Amazon	3604	4000	134689	9.24×10^{-3}
RYM	33786	5381	613387	3.37×10^{-3}

from 1 to 5 stars in MovieLens and Amazon and from 1 to 10 in RYM. A higher rating conveys stronger confidence for user's preference towards an object. Here we only consider objects collected by users with ratings of at least 3 stars in MovieLens and Amazon and no less than 5 stars in RYM for the sake of capturing user's preference more precisely. After processing, detailed information of the datasets is shown in Table 1. Before the experiments, datasets are randomly divided into two parts: a training set E^T containing 90% of all links and a probe set E^P containing the rest.

4.2. Evaluation Metrics. Accuracy is critical in evaluating the performance of the algorithm. We introduce four indicators to assess algorithm's accuracy.

(1) *Averaged Ranking Score ($\langle r \rangle$)* [25]. $\langle r \rangle$ evaluates the ability of ranking users' preferable objects in higher places than disliked ones. For an arbitrary link l_{ij} in E_p , if o_j 's rank in u_i 's recommendation list is rank_{ij} , then the averaged ranking score is defined as

$$\langle r \rangle = \frac{1}{|E_p|} \sum_{l_{ij} \in E_p} \frac{\text{rank}_{ij}}{n - k(u_i)}, \quad (4)$$

where $|E_p|$ denotes the number of links in probe set. The smaller $\langle r \rangle$, the better the algorithm's accuracy.

(2) *Area under ROC Curve (AUC)* [24]. AUC measures the capacity of identifying the relevant objects from the irrelevant objects. For N independent experiments, each of which compares a relevant object and an irrelevant one, if there are n_1 times when the relevant object has a higher score than the irrelevant one and n_2 times when the scores are equal, then

$$\text{AUC} = \frac{n_1 + 0.5n_2}{N}. \quad (5)$$

Clearly, the greater the AUC, the higher the algorithm's accuracy.

(3) *Precision (P)* [13]. Precision is the ratio of the number of u_i 's hidden links (objects collected by u_i and present in the probe set) to $T_i(L)$, contained in the top- L recommendation list. Therefore, the precision P of the whole system is

$$P = \frac{1}{m} \sum_{i=1}^m \frac{T_i(L)}{L}. \quad (6)$$

(4) *Recall* [24]. Recall is the proportion of the number of all hitting links and the number of links in probe set, as

$$\text{Recall}(L) = \frac{1}{|E_p|} \sum_{i=1}^m T_i(L). \quad (7)$$

Diversity quantifies how different the recommended objects are with respect to each other. It is mainly measured from two aspects in this paper.

(1) *Intrasimilarity (I)* [24]. I measures the diversity of the objects in one user's recommendation list. For an arbitrary user, denote the recommended objects as $L_i = \{o_1, o_2, \dots, o_L\}$. Then the whole system's intrasimilarity is written as

$$I = \frac{1}{mL(L-1)} \sum_{l=1}^m \sum_{o_i, o_j \in E_p, i \neq j} s_{ij}^o, \quad (8)$$

where s_{ij}^o is the cosine similarity between o_i and o_j and it is defined as

$$s_{ij}^o = \frac{1}{\sqrt{k(o_i)k(o_j)}} \sum_{l=1}^m a_{il}a_{lj}. \quad (9)$$

The lower the intrasimilarity, the higher the algorithm's diversity.

(2) *Hamming Distance (H)* [29]. H refers to how different the recommended lists are between users. Let D_{ij} denote the number of distinct objects in the recommendation lists of u_i and u_j , and then the averaged hamming distance is

$$H = \frac{1}{m(m-1)} \sum_{i \neq j} \frac{D_{ij}}{L}. \quad (10)$$

The larger H , the higher the algorithm's diversity.

Novelty is closely related to personality which measures the capacity of generating novel and niche recommendations.

(1) *Average Degree ($\langle k \rangle$)* [29]. Let o_{ij} signify the j th recommended object for u_i , and then the average degree of all recommended objects is equal to

$$\langle k \rangle = \frac{1}{mL} \sum_{i=1}^m \sum_{j=1}^L k(o_{ij}). \quad (11)$$

The smaller $\langle k \rangle$, the higher the algorithm's novelty.

4.3. Benchmark Methods. For comparison, we briefly introduce six recommendation algorithms.

(1) *Collaborative Filtering (CF)* [12]. CF is based on measuring the similarity between users or objects. For any two users u_i and u_j , the cosine similarity is

$$s_{ij} = \frac{1}{\sqrt{k(u_i)k(u_j)}} \sum_{l=1}^n a_{il}a_{jl}. \quad (12)$$

Thus the extent to which the target u_i will like o_j is

$$v_{ij} = \frac{\sum_{l=1, l \neq i}^m s_{li}a_{lj}}{\sum_{l=1, l \neq i}^m s_{li}}. \quad (13)$$

(2) *Network-Based Inference (NBI)* [13]. NBI have been introduced before.

(3) *Heterogenous Network-Based Inference (HNBI)* [16]. HNBI is a modified NBI dependent on the initial resource configuration with weight:

$$w_{ij}^{\text{HNBI}} = [k(o_j)]^\beta w_{ij}, \quad (14)$$

where w_{ij} is achieved via (1).

(4) *Corrected Similarity-Based Inference (CSI)* [24]. CSI combines the forward and backward diffusion to correct similarity estimation deviation. Then the resource transfer weight is defined as

$$w_{ij}^{\text{CSI}} = \sqrt{w_{ij} \times \frac{w_{ji}}{\sum_{l=1}^n w_{li}}}, \quad (15)$$

where w_{ij} is achieved via (1).

(5) *Redundant Eliminated Network-Based Inference (RENBI)* [20]. RENBI considers high order correlations between objects, and then the transfer weight is

$$W^{\text{RENBI}} = W + \alpha W^2, \quad (16)$$

where $\alpha \leq 0$ is a free parameter and W is obtained according to (1).

(6) *Preferential Diffusion (PD)* [27]. PD's transfer weight w_{ij} is equal to (2).

Although RENBI and PD are similar to our proposed algorithm, they do not have a complete understanding of the problem of excessive diffusion, so they can only partially alleviate it. SED cleverly combines the advantages of RENBI and PD, which penalizes the popularity degree of objects to relieve popularity bias and further reduces redundant similarities, and can be more effective in suppressing excessive diffusion.

5. Results

In order to obtain credible experimental results, 10-fold cross-validation is performed to decrease deviation. Results

presented in here are achieved via averaging over 10 independent E^T/E^P divisions. The recommendation performances, measured by seven metrics, of seven methods for three datasets are summarized in Table 2.

As shown in Table 2, for the three datasets, SED performs the best on all seven metrics. Concretely speaking, SED surpasses the original mass diffusion-based algorithm NBI in all aspects, especially with $\langle r \rangle$ reduced by 23.6%, P increased by 24.3%, Recall increased by 24.3%, H increased by 37.6%, and $\langle k \rangle$ reduced by 32.8% in MovieLens; Recall increased by 20.1% and $\langle k \rangle$ reduced by 50% in Amazon; and $\langle r \rangle$ reduced by 40.6%, P increased by 25.8%, Recall increased by 26.4%, and $\langle k \rangle$ reduced by 40.8% in RYM. It achieves different degrees of improvement on all other algorithms as well. In a word, the proposed algorithm exhibits outstanding accuracy, diversity, and novelty.

To explain our algorithm's performance under different recommendation list length, we fix the optimal parameters and then vary L from 5 to 100 to obtain the recommendation performance for three datasets, the results of which are shown in Figures 3, 4, and 5, respectively. Since $\langle r \rangle$ and AUC are constant values and do not change with L , they are not shown in Figures 3–5. Each data point is obtained by averaging over ten independent runs with data division.

From the three figures, we can see that for two “the smaller the better” metrics, I and $\langle k \rangle$, SED's curves are at the bottom, while, for the remaining three “the higher the better” metrics, they are always on the top. That is to say, the proposed method also performs best among the schemes under different recommendation list lengths which further supports the results reported in Table 2. From this we can draw a conclusion that the moderate inhibition of excessive diffusion to ensure the fairness of diffusion is more conducive to effective recommendation.

6. Conclusion and Discussion

Motivated by preventing the resources excessively diffused to the popular objects and the objects where there exists redundant similarity, in this paper, we firstly penalize the popular objects' degrees and take the heterogeneity of users' degrees into account simultaneously to restrain popularity bias and then we eliminate redundant similarities to some extent by subtracting the second-order similarity. Extensive experiments on three real datasets consistently demonstrate the effectiveness of SED considering its improvement in accuracy, diversity, and novelty. SED also performs the best compared to the benchmarks under different recommendation list lengths, and thus it is applicable and versatile in practice.

Our method can more accurately match the user with the right objects complying with his/her preferences and in a commercial sense can further grasp user's loyalty to promote substantial profits growth. Because of its effectiveness, SED can be applied in various kinds of recommendation environments, like using purchase records to recommend books, using reading histories to recommend news, recommending TV shows and movies on the basis of users' viewing patterns

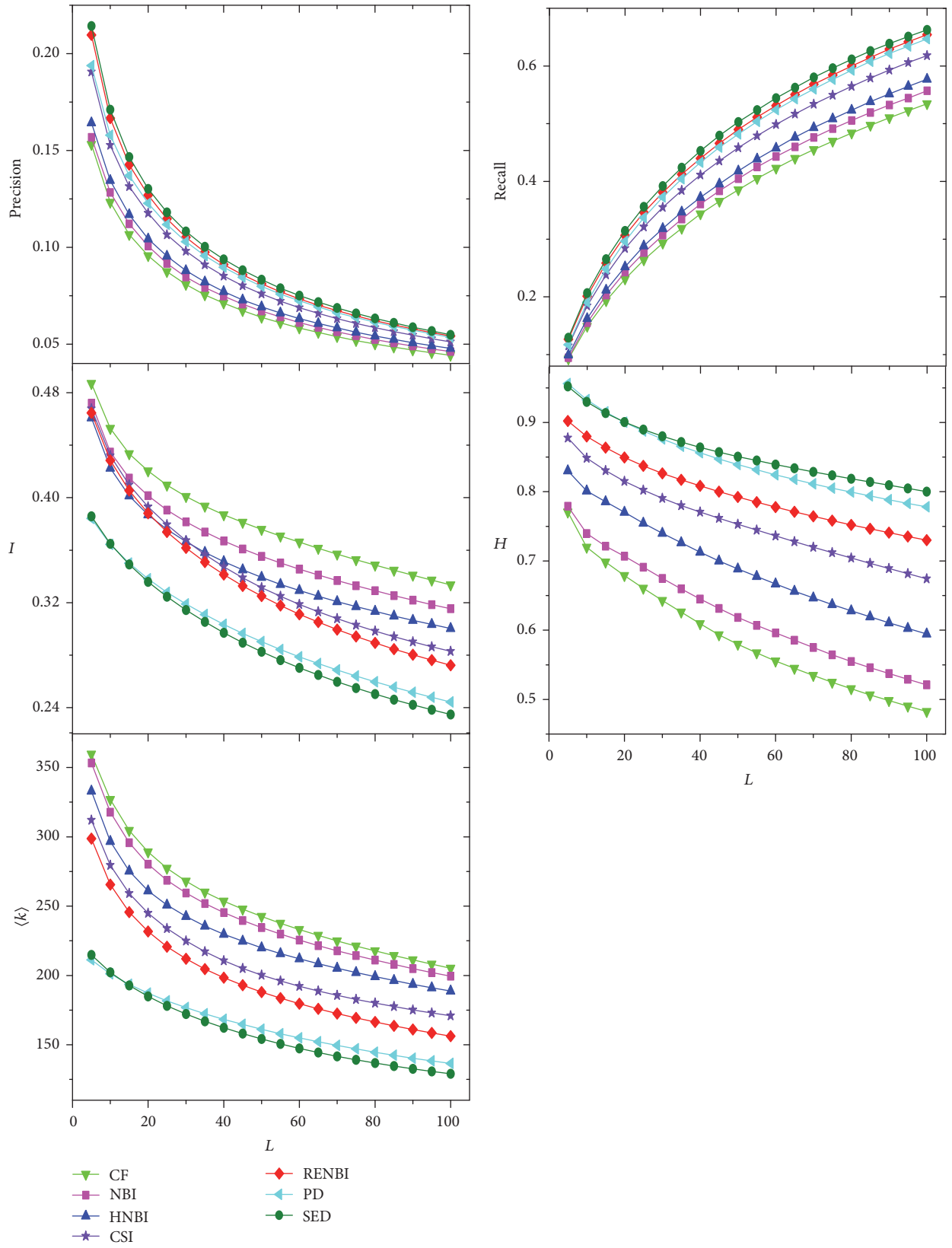


FIGURE 3: Algorithms' performance under different recommendation lengths for MovieLens.

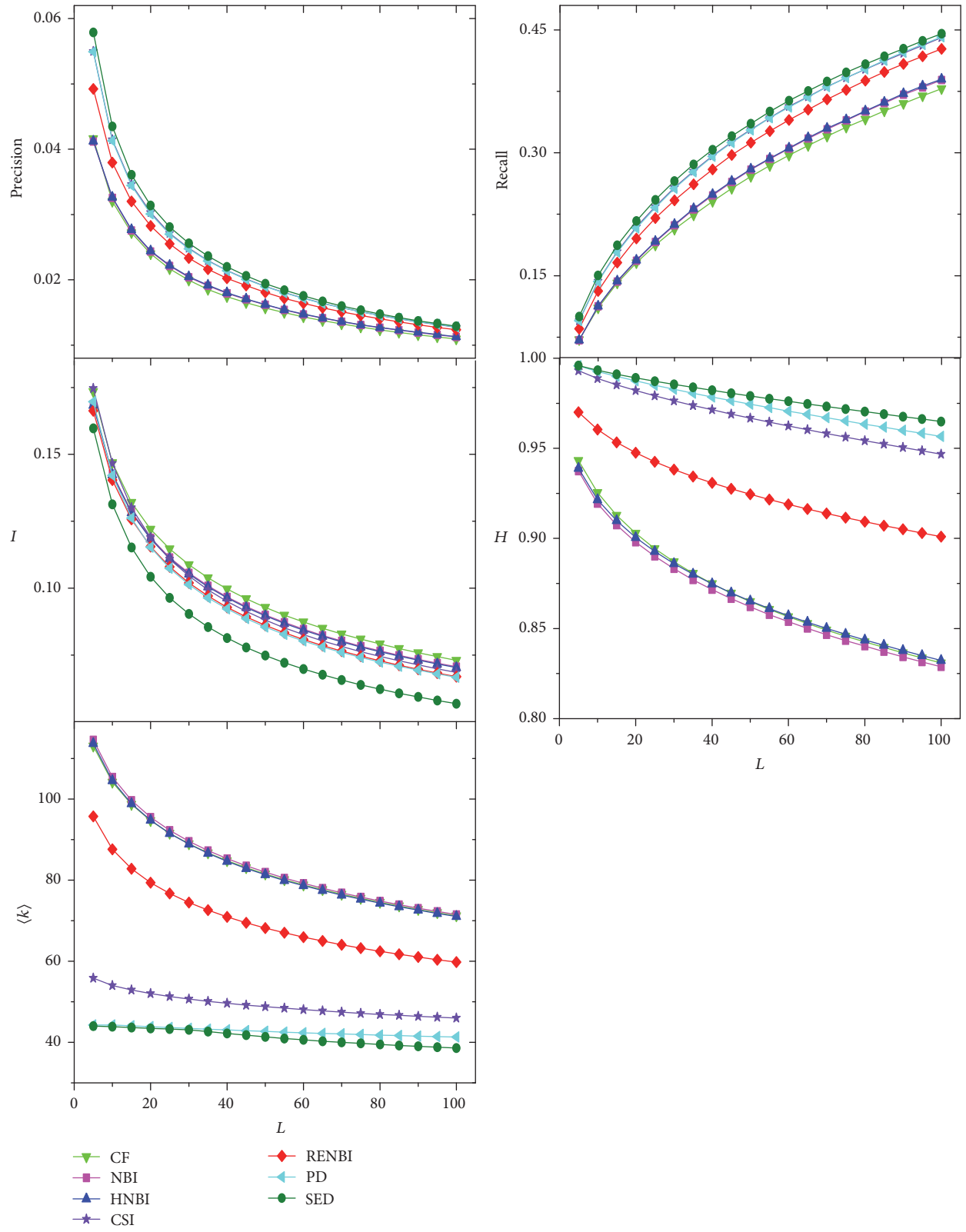


FIGURE 4: Algorithms' performance under different recommendation lengths for Amazon.

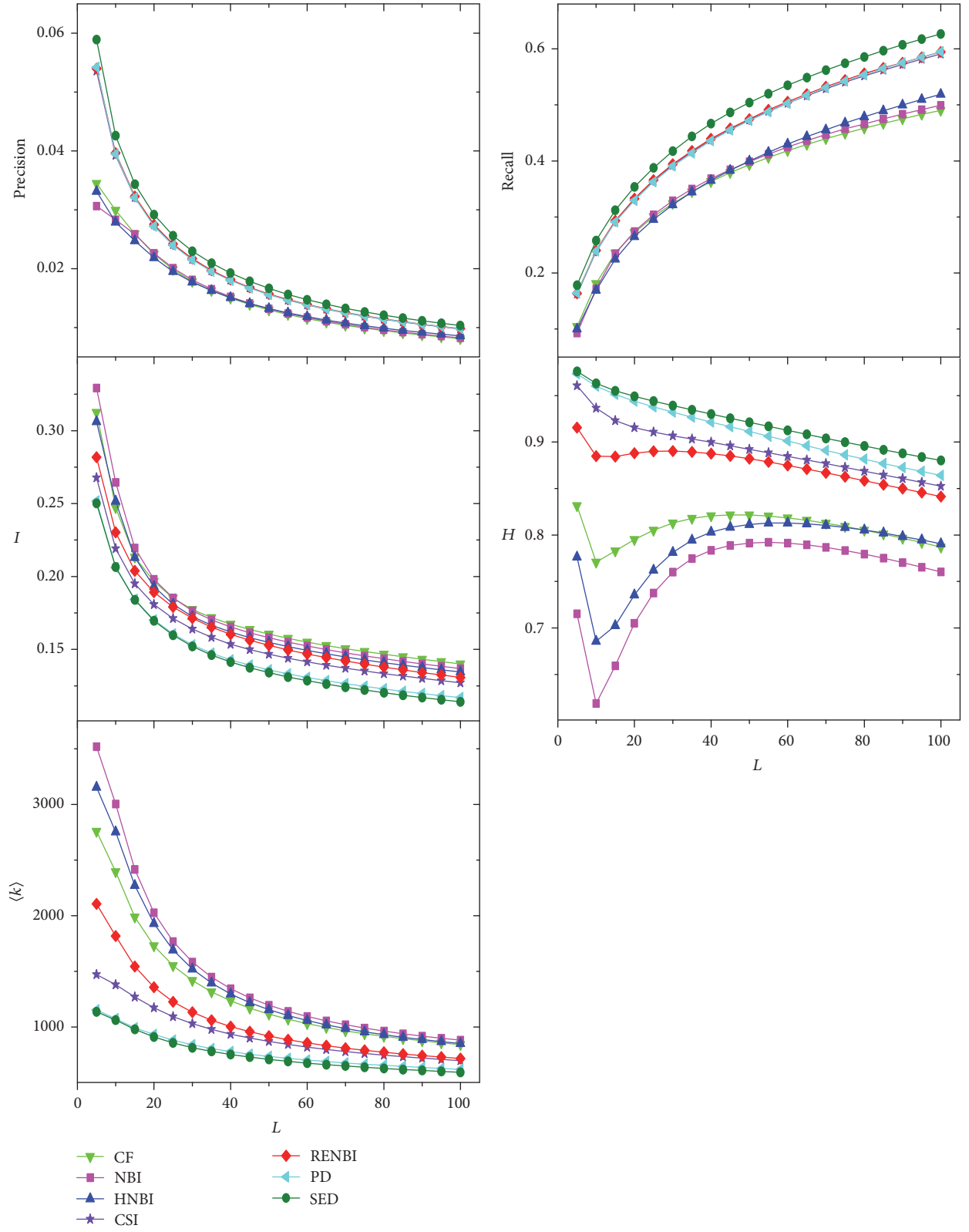


FIGURE 5: Algorithms' performance under different recommendation lengths for RYM.

TABLE 2: Algorithms' performances on three datasets with $L = 50$. For fair comparison with parameter-free algorithms, the corresponding parameters which achieve the lowest ranking score are chosen with resolution of 0.01 ($\beta_{\text{opt}} = -0.86$ for HNBI, $\alpha_{\text{opt}} = -0.76$ for RENBI, $\beta_{\text{opt}} = -0.83$ for PD, and $\alpha_{\text{opt}} = -0.85$ and $\beta_{\text{opt}} = -0.55$ for SED in MovieLens; $\beta_{\text{opt}} = -0.08$ for HNBI, $\alpha_{\text{opt}} = -0.53$ for RENBI, $\beta_{\text{opt}} = -0.62$ for PD, and $\alpha_{\text{opt}} = -0.89$ and $\beta_{\text{opt}} = -0.36$ for SED in Amazon; and $\beta_{\text{opt}} = -0.77$ for HNBI, $\alpha_{\text{opt}} = -0.67$ for RENBI, $\beta_{\text{opt}} = -0.77$ for PD, and $\alpha_{\text{opt}} = -0.69$ and $\beta_{\text{opt}} = -0.67$ for SED in RYM). The sampling number N in AUC is fixed as 10^6 . The best values of each metric are emphasized in boldface.

MovieLens	$\langle r \rangle$	AUC	P	Recall	I	H	$\langle k \rangle$
CF	0.1225	0.8990	0.0639	0.3856	0.3758	0.5796	243
NBI	0.1143	0.9094	0.0670	0.4048	0.3554	0.6185	235
HNBI	0.1075	0.9144	0.0693	0.4183	0.3392	0.6886	220
CSI	0.0970	0.9278	0.0759	0.4584	0.3315	0.7530	200
RENBI	0.0875	0.9349	0.0812	0.4901	0.3250	0.7923	188
PD	0.0877	0.9341	0.0798	0.4819	0.2902	0.8392	161
SED	0.0873	0.9352	0.0833	0.5031	0.2895	0.8508	158
Amazon	$\langle r \rangle$	AUC	P	Recall	I	H	$\langle k \rangle$
CF	0.1212	0.8811	0.0157	0.2711	0.0928	0.8650	82
NBI	0.1170	0.8844	0.0162	0.2795	0.0900	0.8620	82
HNBI	0.1169	0.8843	0.0162	0.2803	0.0896	0.8653	81
CSI	0.1036	0.8936	0.0190	0.3284	0.0881	0.9667	49
RENBI	0.1103	0.8848	0.0181	0.3124	0.0861	0.9245	68
PD	0.1031	0.8935	0.0190	0.3277	0.0855	0.9745	43
SED	0.1024	0.8962	0.0194	0.3356	0.0753	0.9782	41
RYM	$\langle r \rangle$	AUC	P	Recall	I	H	$\langle k \rangle$
CF	0.0756	0.9548	0.0130	0.3932	0.1605	0.8216	1114
NBI	0.0675	0.9612	0.0132	0.3989	0.1580	0.7912	1196
HNBI	0.0588	0.9641	0.0132	0.3997	0.1548	0.8114	1155
CSI	0.0463	0.9715	0.0156	0.4731	0.1467	0.8922	869
RENBI	0.0456	0.9701	0.0157	0.4748	0.1527	0.8820	918
PD	0.0440	0.9719	0.0156	0.4719	0.1359	0.9114	734
SED	0.0401	0.9720	0.0166	0.5042	0.1340	0.9214	708

and ratings, and so on. Although SED has achieved good recommendation performance, there are some improvements worthy of further investigation, such as considering the time dimension, which is indeed necessary as it can greatly affect the recommendation performance and measure the methods ability to reflect both network topology and the systems natural growth patterns and the users shifting interests [30]. In summary, we hope our method can enlighten readers to a certain extent.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

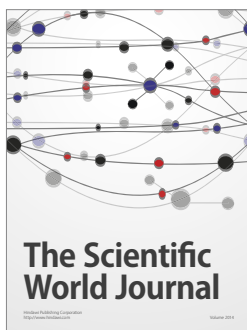
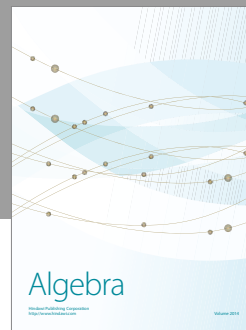
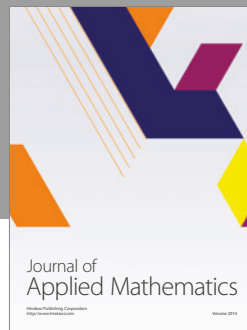
Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61602048 and 61471060) and the Project Funded by China Postdoctoral Science Foundation (no. 2015M581030).

References

- [1] S. Goswami, "Analysing effects of information overload on decision quality in an online environment," *Journal of Management Research*, vol. 15, no. 4, pp. 231–245, 2015.
- [2] R. T. B. Ma, J. Lui, and V. Misra, "Evolution of the internet economic ecosystem," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 1, pp. 85–98, 2015.
- [3] G.-D. Xu, Z. Wu, Y. Zhang, and J. Cao, "Social networking meets recommender systems: survey," *International Journal of Social Network Mining*, vol. 2, no. 1, pp. 64–100, 2015.
- [4] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*, Springer US, 2011.
- [5] D. Bernardes, M. Diaby, R. Fournier, F. Fogelman-Soulié, and E. Viennet, "A social formalism and survey for recommender systems," *ACM SIGKDD Explorations Newsletter*, vol. 16, no. 2, pp. 20–37, 2015.
- [6] J. Lu, D. S. Wu, M. S. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.
- [7] N.-S. Bhosale and S.-S. Pande, "A survey on recommendation system for big data applications," *Data Mining and Knowledge Engineering*, vol. 7, no. 1, pp. 42–44, 2015.

- [8] C. He, D. Parra, and K. Verbert, "Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities," *Expert Systems with Applications*, vol. 56, pp. 9–27, 2016.
- [9] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," in *Recommender Systems Handbook*, pp. 1–34, Springer US, 2015.
- [10] L. Sharma and A. Gera, "A survey of recommendation system: research challenges," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, no. 5, pp. 1989–1992, 2013.
- [11] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," *Computer Communications*, vol. 41, pp. 1–10, 2014.
- [12] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, article 3, Article ID 2556270, 2014.
- [13] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, Article ID 046115, 2007.
- [14] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, and F. Yang, "Recommendation model based on opinion diffusion," *Europhysics Letters. EPL*, vol. 80, no. 6, Article ID 68003, Art. 68003, 5 pages, 2007.
- [15] A. Fiasconaro, M. Tumminello, V. Nicosia, V. Latora, and R. N. Mantegna, "Hybrid recommendation methods in complex networks," *Physical Review E*, vol. 92, no. 1, Article ID 012811, 2015.
- [16] T. Zhou, L. L. Jiang, R.-Q. Su, and Y.-C. Zhang, "Effect of initial configuration on network-based recommendation," *Europhysics Letters. EPL*, vol. 81, no. 5, Article ID 58004, 2008.
- [17] D.-C. Nie, Y. Fu, J.-L. Zhou, Z. Liu, Z.-K. Zhang, and C. Liu, "A personalized recommendation algorithm via biased random walk," in *Proceedings of the 11th International Joint Conference on Computer Science and Software Engineering. (JCSSE '14)*, pp. 292–296, IEEE, Chon Buri, Thailand, 2014.
- [18] Y.-H. An, Q. Dong, C.-J. Sun, D.-C. Nie, and Y. Fu, "Diffusion-like recommendation with enhanced similarity of objects," *Physica A: Statistical Mechanics and Its Applications*, pp. 1–8, 2016.
- [19] 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.
- [20] T. Zhou, R.-Q. Su, R.-R. Liu, L.-L. Jiang, B.-H. Wang, and Y.-C. Zhang, "Accurate and diverse recommendations via eliminating redundant correlations," *New Journal of Physics*, vol. 11, no. 12, Article ID 123008, 2009.
- [21] J.-G. Liu, T. Zhou, and Q. Guo, "Information filtering via biased heat conduction," *Physical Review E*, vol. 84, no. 3, Article ID 037101, 2011.
- [22] W.-P. Ma, X. Feng, S. Wang, and M. Gong, "Personalized recommendation based on heat bidirectional transfer," *Physica A: Statistical Mechanics and Its Applications*, vol. 444, pp. 713–721, 2016.
- [23] B. Zhang, S. Tang, Z. Ying, Y. Cai, G. Xu, and K. Xu, "A novel recommendation algorithm based on heterogeneous information network similarity and preference diffusion," in *International Conference on Web-Age Information Management*, pp. 53–64, Springer International Publishing, 2015.
- [24] X.-Z. Zhu, H. Tian, and S.-M. Cai, "Personalized recommendation with corrected similarity," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2014, no. 7, Article ID P07004, 2014.
- [25] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. 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.
- [26] D.-C. Nie, Y.-H. An, Q. Dong, Y. Fu, and T. Zhou, "Information filtering via balanced diffusion on bipartite networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 421, pp. 44–53, 2015.
- [27] L.-Y. Lü and W.-P. Liu, "Information filtering via preferential diffusion," *Physical Review E*, vol. 83, no. 6, 2011.
- [28] A. Zeng, A. Vidmer, M. Medo, and Y.-C. Zhang, "Information filtering by similarity-preferential diffusion processes," *Europhysics Letters*, vol. 105, no. 5, Article ID e58002, 2014.
- [29] X. Zhu, H. Tian, P. Zhang, Z. Hu, and T. Zhou, "Personalized recommendation based on unbiased consistence," *EPL (Europhysics Letters)*, vol. 111, no. 4, Article ID 48007, 2015.
- [30] A. Vidmer and M. Medo, "The essential role of time in network-based recommendation," *Europhysics Letters*, vol. 116, no. 3, Article ID 30007, 2016.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

