

Balancing the popularity bias of object similarities for personalised recommendation

Lei Hou^{1,a}, Xue Pan¹, and Kecheng Liu^{1,2}

¹ Informatics Research Centre, Henley Business School, University of Reading, Reading, Berkshire RG6 6UD, UK

² Data Science and Cloud Service Research Centre, Shanghai University of Finance and Economics, Shanghai 200433, P.R. China

Received 27 June 2017 / Received in final form 18 October 2017

Published online 7 March 2018

© The Author(s) 2018. This article is published with open access at [Springerlink.com](https://www.springerlink.com)

Abstract. Network-based similarity measures have found wide applications in recommendation algorithms and made significant contributions for uncovering users' potential interests. However, existing measures are generally biased in terms of popularity, that the popular objects tend to have more common neighbours with others and thus are considered more similar to others. Such popularity bias of similarity quantification will result in the biased recommendations, with either poor accuracy or poor diversity. Based on the bipartite network modelling of the user-object interactions, this paper firstly calculates the expected number of common neighbours of two objects with given popularities in random networks. A Balanced Common Neighbour similarity index is accordingly developed by removing the random-driven common neighbours, estimated as the expected number, from the total number. Recommendation experiments in three data sets show that balancing the popularity bias in a certain degree can significantly improve the recommendations' accuracy and diversity simultaneously.

1 Introduction

The overwhelming online information, though provides users massive and diverse choices, is making it more and more difficult to find what they really want. Accordingly, users largely rely on the information filtering systems, such as the search engines and recommender systems [1–3], to look for relevant information.

The recommender systems have got significant attentions and wide applications over the past decades due to its advances in finding users' potential interests [4–6]. Various algorithms have been developed including the content-based systems which applies the object information such as attributes [7], contents [8] and tags [9,10] to define similarities, and the most widely used collaborative filtering systems [1,11,12]. Modelling the interactions between users and objects as bipartite networks [13,14], collaborative filtering systems normally examine the association patterns of either users accessing same objects, or objects being accessed by the same users, leading to the user-based and object-based collaborative filtering respectively. Many practical online systems are object-based systems using object similarities, such as the Amazon's product recommender system [4] and YouTube's video recommender system [5]. These systems generally recommend objects that are similar to the target users' historical

selections. The quantification of object similarity consequently becomes the crucial part for such kind of recommender systems.

Based on the user-object bipartite network modelling, where the users and objects are abstracted as two sets of vertices, the object similarities can be evaluated according to the structure of the network. The most widely-used such approach is known as the Common Neighbour (CN) index which examines the size of two objects' common neighbourhood. If letting Γ_α be the set of users who are connecting to object α , the CN similarity between objects α and β can be described as $s_{\alpha\beta}^{CN} = |\Gamma_\alpha \cap \Gamma_\beta|$. Considering the popularities, i.e. degrees, of the objects k , a number of variations have been developed, such as the Leicht-Holme-Newman (LHN) index [15] and the Hub-Promoted (HP) index [16], which read $s_{\alpha\beta}^{LHN} = |\Gamma_\alpha \cap \Gamma_\beta|/k_\alpha k_\beta$ and $s_{\alpha\beta}^{HP} = |\Gamma_\alpha \cap \Gamma_\beta|/\min(k_\alpha, k_\beta)$ respectively. On the other hand, the Adamic-Adar (AA) index [17] and Resource Allocation (RA) index [18] take the user degree as weight to the CN index and thus read $s_{\alpha\beta}^{AA} = \sum_{i \in \Gamma_\alpha \cap \Gamma_\beta} 1/\log(k_i)$ and $s_{\alpha\beta}^{RA} = \sum_{i \in \Gamma_\alpha \cap \Gamma_\beta} 1/k_i$ respectively. Some diffusion processes have also been introduced to measure the vertex similarity in bipartite networks, and the Mass Diffusion (MD) index [13] and Heat Conduction (HC) index [19] are proposed which read $s_{\alpha\beta}^{MD} = \frac{1}{k_\alpha} \sum_{i \in \Gamma_\alpha \cap \Gamma_\beta} 1/k_i$ and $s_{\alpha\beta}^{HC} = \frac{1}{k_\beta} \sum_{i \in \Gamma_\alpha \cap \Gamma_\beta} 1/k_i$ respectively. For a review on these similarity quantifications, see reference [3].

^a e-mail: l.hou@pgr.reading.ac.uk

Though have been widely applied in both analytics of complex networks and the recommender systems, most of the existing similarity indices have systematic popularity bias [12,20,21]. Objects, in general, would tend to be more similar to either very popular objects (for indices such as CN and MD) or very unpopular objects (for indices such as HC). As a consequence, the recommendations would be biased in terms of the popularity leading to either poor accuracy (HC) or poor diversity (CN, MD). Additionally, given the scope of serving online users, both the accuracy and diversity have been argued to be crucial for recommender systems [22–24]. Therefore, to balance the popularity bias of object similarities and explore to what extent should the system recommend popular objects is significant for achieving accurate and diverse recommendations.

In this paper we develop a Balanced Common Neighbour (BCN) index for measuring the object similarity in bipartite networks based on the evaluation of the expected common neighbourhood for two objects with specific popularities. Applying the proposed method in personalised recommendation, we show that to balance the popularity bias of similarities in a certain degree can largely improve the performances of the recommendations.

2 Balanced object similarity index

The most fundamental CN index regards two objects that have been collected by many same users as similar to each other. However, popular objects which have been collected by a large number of users tend to have more common neighbours (be more similar) with others than those unpopular ones. As shown in Figure 1, the popular objects are frequently evaluated by the CN index as the most similar objects to others. If ranking randomly, the popularity distribution of the top similar objects should be expected to be same as the object popularity distribution of the empirical data. However, the distribution of CN top similar objects is much higher than the empirical distribution on the tail (the range of popular objects). As a consequence, the recommender system has a tendency to recommend popular objects to users, which though may lead to good accuracy, will result in bad personalisation.

Actually, the number of common neighbours for two objects α and β consists of two components that one comes from random mechanism $n_{\alpha\beta}^{rand}$ and the other comes from the similarity between them $n_{\alpha\beta}^{sim}$. While the random component is completely popularity-correlated, the other one describes purely the similarity regardless of their popularities. To distinguish these two components is thus of significance for us to control the popularity tendency and the similarity tendency of the recommender system to optimise the recommendations. To do so, we firstly assume the user-object bipartite network to be completely random and explore the expected number of common neighbours $n_{\alpha\beta}^{exp}$ for two arbitrary objects with given popularities.

Considering a random user-object bipartite network with N users and M objects, we let T to be the total links between users and objects, i.e. $T = \sum_u k_u = \sum_o k_o$. We let all the links between users and objects break into

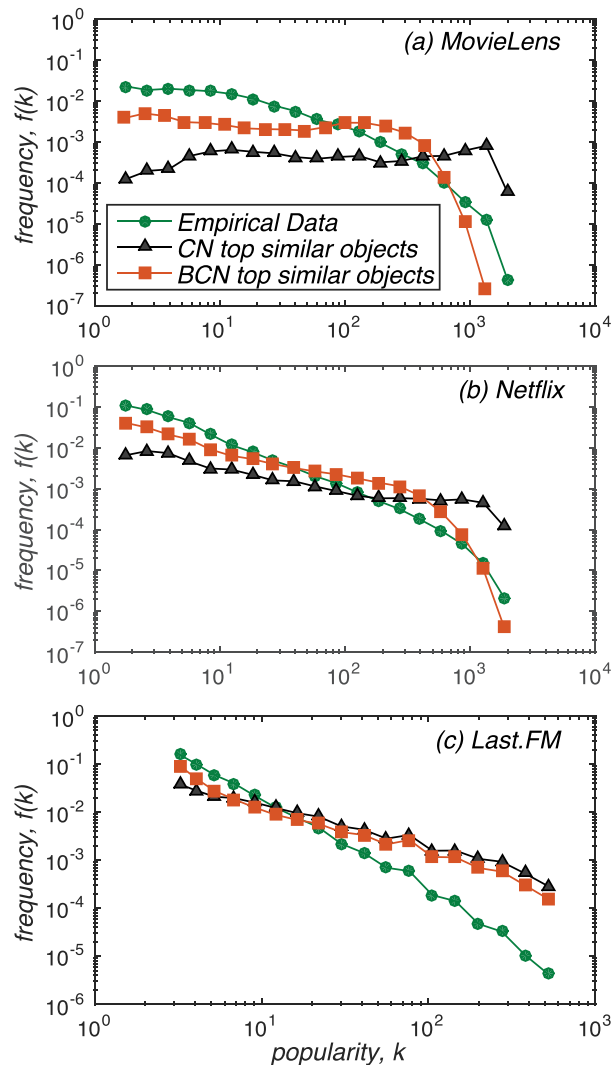


Fig. 1. Distributions of object popularity (degree) in datasets (a) MovieLens, (b) Netflix and (c) Last.FM respectively. The introduction and basic statistics of these datasets can be found in Table 1. In each of the subplots, the green circles are the distribution of all the objects, i.e. the basic degree distribution of the object side of the user-object bipartite network. For each object, we calculate the similarities of it with each of the others using the CN index and proposed BCN index with parameter $\lambda = 1$, and the corresponding curves (black triangles for CN index and red square for BCN index) show the distributions of popularities of the top-20 similar objects.

half-links so that we can explore the expected number of common neighbours in a process of random rewiring, which is also known as the configuration model [25–27]. Assuming an object β has randomly connected to k_β users, and we let T_β to be the total number of remaining half-links originating from these k_β users. For each half-link of β 's, its probability to connect to a user u can be given by k_u/T , leading the expected degree of the user at the end of each link of β 's to be $\sum_u k_u^2/T$. Accordingly, we have,

$$T_\beta = k_\beta \cdot \left(\sum_u k_u^2/T - 1 \right). \quad (1)$$

The number of common neighbours between any other object α with β is thus determined by the process where α select k_α out of $T - k_\beta$ half-links. When one of the T_β half-links is selected, one CN is generated for α and β . Therefore, the number of common neighbours between α and β , $n_{\alpha\beta}$ can be described by an hypergeometric distribution $H(n_{\alpha\beta}; k_\alpha, T_\beta, T - k_\beta)$. For any bipartite network which is sparse enough, i.e. $T \gg k_o, \forall o$, we can approximately have $H(n_{\alpha\beta}; k_\alpha, T_\beta, T)$ to describe the distribution of $n_{\alpha\beta}$. The mean of such hypergeometric distribution is thus the expected number of common neighbours between two random objects, which reads

$$n_{\alpha\beta}^{exp} = \frac{k_\alpha k_\beta T_\beta}{T} = \frac{\langle k_u^2 \rangle - \langle k_u \rangle}{N \langle k_u \rangle^2} \cdot k_\alpha k_\beta. \quad (2)$$

Note that, such consideration does not exclude the case of multi-links, leading the calculated value slightly higher than the actual theoretical value for number of common neighbours, especially for those objects with very large degrees. However, the expression is valid for the sparse limit or the limit of $N \rightarrow \infty; M \rightarrow \infty$.

The derivation of the expected number of common neighbours is based on a null model of bipartite networks where the two kinds of nodes connects to each other randomly, and thus the resulted expression is similar to the ones in other similar context [15,28]. As shown by equation (2), the expected number of common neighbours between two objects α and β is linearly correlated with the product of their popularities $k_\alpha k_\beta$. We further use \mathcal{H} to denote the parameter before the product of the popularities, and it can be rewritten as

$$\mathcal{H} = \frac{\sum_u k_u^2 - \sum_u k_u}{(\sum_u k_u)^2} = \frac{1}{N} \left(\frac{\langle k_u^2 \rangle}{\langle k_u \rangle^2} - \frac{1}{\langle k_u \rangle} \right), \quad (3)$$

where $\langle \cdot \rangle$ represents the mean value. In the parameter, the component $\langle k_u^2 \rangle / \langle k_u \rangle^2$ is normally referred as the degree heterogeneity H of a network [18,29]. A large value of H may suggest that the network's degree distribution is very heterogeneous. As a consequence, the parameter \mathcal{H} describes the heterogeneity of user degree distribution.

With the expected number of common neighbours as the estimation for the random component, one can compare the actual and expected number by taking either ratios or differences to get the similarity index. Normally, real user-object systems are extremely sparse where most object pairs would have no common neighbours at all. In order to make these object pairs distinguishable from each other, we define the similarity between two objects α and β by taking difference as $n_{\alpha\beta}^{sim} = s_{\alpha\beta}^{CN} - n_{\alpha\beta}^{exp} = |\Gamma_\alpha \cap \Gamma_\beta| - \mathcal{H} k_\alpha k_\beta$. This expression can be used as an object similarity index and theoretically there would be no popularity bias for quantified object similarities. Considering the popularity of objects may be an influential factor in recommender systems, this paper explores to what extent should the popularity bias be balanced to achieve good performance. By introducing a free parameter λ , we define a new similarity index, namely the BCN

Table 1. Statistics of the applied datasets. The MovieLens and Netflix datasets are records of users watching movies and the Last.FM dataset is the records that users chose from a group of artists to follow. All of these three datasets can be modelled as the user-object bipartite networks and have been widely used in the studies and tests of recommendation algorithms. In the table, N , M and T represent the number of users, objects and total links respectively, H is the user degree heterogeneity calculated as $\langle k_u^2 \rangle / \langle k_u \rangle^2$, and \mathcal{H} is the heterogeneity parameter defined in equation (3).

Dataset	N	M	T	H	\mathcal{H}
MovieLens	5547	5850	698054	3.2	5.7×10^{-4}
Netflix	8609	5081	419247	3.7	4.2×10^{-4}
Last.FM	1885	6953	82155	1.0	5.4×10^{-4}

index as

$$s_{\alpha\beta}^{BCN} = |\Gamma_\alpha \cap \Gamma_\beta| - \lambda \cdot \mathcal{H} k_\alpha k_\beta. \quad (4)$$

One may find from the expression that $\lambda = 0$ gives the standard CN index and $\lambda = 1$ gives the theoretical similarity with no popularity bias.

For the theoretical similarity (the BCN index with $\lambda = 1$), the popularity distribution of the top similar objects differs from that of the CN index, as shown in Figure 1. The frequency of the extreme popular objects being considered as most similar to others is much lower than that of the CN index. Instead, the BCN index puts more focus on the middle range, that the objects that are neither extreme popular nor unpopular are more frequently considered as similar to others than the empirical distribution. However, the difference between the distributions of BCN and CN index is relatively small for the Last.FM dataset. This may be due to the fact that the user degree heterogeneity H is almost 1 (Tab. 1) in Last.FM, while that of the MovieLens and Netflix datasets are 3.2 and 3.7 respectively.

3 Recommendation algorithm and evaluation metrics

A recommender system can normally be described as a user-object bipartite network with a set of users $U = \{u_1, u_2, \dots, u_N\}$, and a set of objects $O = \{o_1, o_2, \dots, o_M\}$. The interactions between users and objects can thus be presented by the adjacency matrix $A = \{a_{uo}\}$ where $a_{uo} = 1$ if the user u collected the object o , and 0 otherwise. Therefore, the task of the recommender systems is actually to predict a number of new links between unconnected user-object pairs. For a target user i , a score for an object α to be connected, $w_{i\alpha}$, can be calculated as,

$$w_{i\alpha} = \sum_{o \in O} a_{io} \cdot s_{o\alpha}, \quad (5)$$

where $s_{o\alpha}$ is the similarity between objects o and α that calculated by an arbitrary index. Note that, this paper focuses on the proposed BCN index, and also considers all the mentioned indices as comparisons. As shown by the equation, the score of object α to be collected is actually the summation of similarities of it to all the target user's historical collections. One can rank the score of all the uncollected objects, and these ranked at the top L positions will be regarded as the recommendation list for the target user.

To evaluate the performance of the recommendations, one can randomly divide the data (links of the bipartite network) into a training set and a probe set. Therefore, one can compare the recommendation lists generated using the training set, with the records in the probe set. This paper considers four widely-used metrics to measure two aspects of the recommendation performances, i.e. the accuracy and diversity.

3.1 Accuracy

For a specific user i , if an object recommended by the system is actually what s/he collected in the probe set, we can call it an accurate recommendation. Denoting the number of accurate recommendations for user i as $h_i(L)$, two accuracy metrics can be defined accordingly, i.e. the Precision and Recall. The precision considers how many of the L recommendations are accurate, while the recall considers how many of the k_i^{probe} removed records are retrieved. Thus, the precision and recall for the target user i are calculated as $p_i(L) = h_i(L)/L$ and $r_i(L) = h_i(L)/k_i^{probe}$ respectively. Averaging over all the users, the precision and recall of the recommender system read

$$P(L) = \frac{1}{N} \sum_{i \in U} \frac{h_i(L)}{L}, \quad (6)$$

and

$$R(L) = \frac{1}{N} \sum_{i \in U} \frac{h_i(L)}{k_i^{probe}}, \quad (7)$$

where U is the set of users. Obviously, both precision and recall are the higher the better with 0 and 1 as the lower- and upper-limits respectively.

3.2 Diversity

It has long been argued that being accurate is not enough for recommender systems [22], because the users may want personalised and novel recommendations. Accordingly, many diversity metrics have been developed. One way to measure the diversity is to evaluate the differences between different users' recommendation lists, which is normally referred as the personalisation. For two users i and j 's recommendation lists, the Hamming distance can be calculated as $ham_{ij} = 1 - Q_{ij}/L$, where Q_{ij} is the number of same objects in the two users' recommendation lists. The personalisation is then defined as the average

Hamming distance over all possible user pairs, i.e.

$$S(L) = \frac{1}{N(N-1)} \sum_{i,j,i \neq j} \left(1 - \frac{Q_{ij}}{L}\right). \quad (8)$$

On the other hand, recommending popular objects to users may be of little value because they can be easily found by users themselves or through other means such as the search engine. Therefore, the novelty of recommendations has also been addressed. While there are several ways of measuring the novelty of recommendations, we follow reference [3] and define the novelty as the average popularity of all the recommended objects, i.e.

$$N(L) = \frac{1}{N \cdot L} \sum_{i \in U} \sum_{\alpha \in \Omega_i} k_{\alpha}, \quad (9)$$

where Ω_i is the set of the L objects that are recommended to user i .

For the personalisation, the higher values may suggest that the recommendations are more personalised. As for the novelty, lower values are expected which may indicate that the recommended objects are novel (unpopular).

4 Results

We carry out recommendation experiments based on three empirical datasets as shown in Table 1. Notably, the user degree heterogeneity H for MovieLens and Netflix is high, while the Last.FM has evenly distributed user degrees ($H \approx 1$). For all the recommendation experiments in this paper, we randomly divide 20% of the links into the probe set for each dataset, and take a recommendation list length $L = 20$. Furthermore, all of the results on the recommendation performance are averaged over 100 independent experiments.

We start with exploring that to what extent should the popularity bias of object similarity should be balanced (by tuning λ) to achieve better recommendation performances in terms of accuracy and diversity. The results are shown in Figure 2. As has been discussed earlier, $\lambda = 0$ gives us the standard CN index, which can result in relatively good accuracy but poor diversity. The reason is that CN recommends generally the extremely popular objects, which have better chance to suite more users' common interests. However, every user's recommendation list would be dominated by the same popular objects, leading to poor personalisation and novelty. When gradually increasing the parameter λ , the BCN index removes more and more randomly-generated common neighbours as suggested by equation (4). Accordingly, the average popularity of the recommended objects, i.e. the novelty $N(20)$, decreases with the increase of λ , as those not-so-popular objects are evaluated as more similar to others as shown in Figure 1. The recommendations thus become more and more personalised. Therefore, Figure 2 suggests that to remove the random component of the common neighbours can remarkably improve the diversity of the

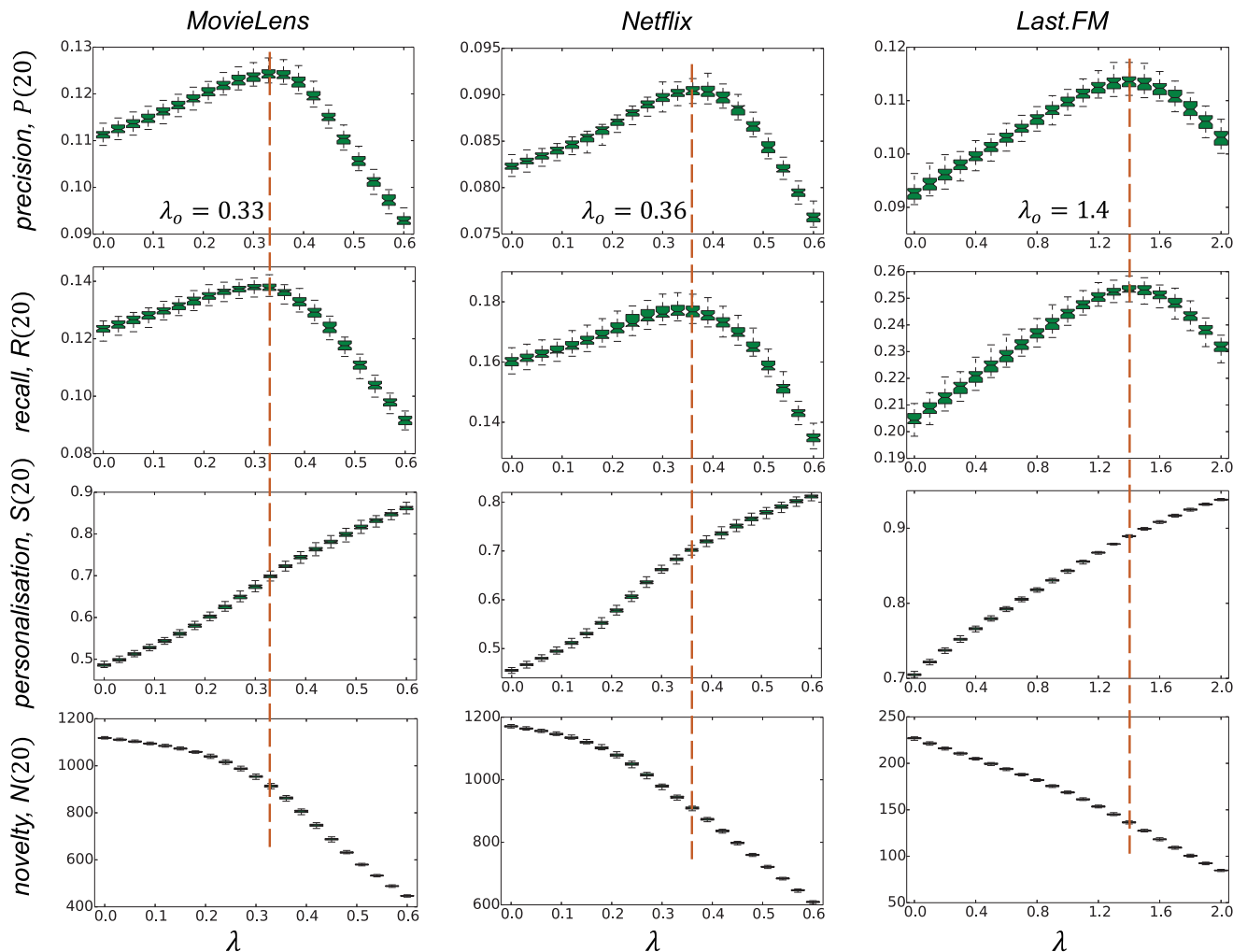


Fig. 2. Recommendation performances on three datasets using BCN similarity index. The precision and recall as the accuracy of the recommendations, are the higher the better. As to the diversity, personalisation is the higher the better while the novelty is the lower the better. Each column of subplots is the result for one dataset, where the red dashed line represent the optimised parameter λ_o maximising the precision. The results for each dataset are based on 100 independent recommendation experiments with random data partitions.

recommendations. On the other hand, the recommendation accuracy will also be largely influenced by the balance of the popularity bias. If slightly balance the popularity bias, the recommendations are shown to be more accurate. However, when applying a large λ , which means to totally remove the popularity bias ($\lambda = 1$) or even reverse the bias ($\lambda > 1$), the recommendation lists would be dominated by only unpopular objects, leading to poor accuracy. Here we take an optimised value λ_o maximising the precision for each dataset. With the optimised value λ_o , both the accuracy and diversity of the recommendations can be significantly improved in comparison to the algorithm applying the original CN index. The optimised values for λ are 0.33, 0.36 and 1.4 for MovieLens, Netflix and Last.FM datasets respectively.

As shown in Table 2, the accuracy metrics (precision $P(20)$ and recall $R(20)$) are improved about 10% for the MovieLens and Netflix datasets and more than 20%

Table 2. Numerical results of recommendation algorithm applying BCN index and its improvements in comparison to the CN index.

	$P(20)$	$R(20)$	$S(20)$	$N(20)$
<i>MovieLens</i>				
$\lambda = 0$	0.111	0.123	0.486	1118.1
$\lambda_o = 0.33$	0.124	0.138	0.699	913.7
Improvement	11.64%	11.68%	43.72%	18.29%
<i>Netflix</i>				
$\lambda = 0$	0.082	0.160	0.455	1171.0
$\lambda_o = 0.36$	0.090	0.177	0.702	909.4
Improvement	9.95%	10.46%	54.37%	22.34%
<i>Last.FM</i>				
$\lambda = 0$	0.093	0.205	0.705	227.0
$\lambda_o = 1.4$	0.114	0.254	0.889	136.5
Improvement	22.52%	23.79%	26.12%	39.89%

Table 3. Comparison of recommendation performances among algorithms applying different similarity indices. The results of BCN index are based on the optimised value of λ_o , i.e. 0.33, 0.36 and 1.4 for the MovieLens, Netflix and Last.FM respectively.

	MovieLens				Netflix				Last.FM			
	$P(20)$	$R(20)$	$S(20)$	$N(20)$	$P(20)$	$R(20)$	$S(20)$	$N(20)$	$P(20)$	$R(20)$	$S(20)$	$N(20)$
CN	0.111	0.123	0.486	1118.1	0.082	0.160	0.455	1171.0	0.093	0.205	0.705	227.0
LHN	0.001	0.001	0.469	1.1	0.001	0.001	0.922	1.4	0.017	0.044	0.994	2.1
HP	0.001	0.001	0.497	1.8	0.001	0.002	0.848	8.7	0.068	0.154	0.948	78.3
AA	0.114	0.130	0.500	1120.7	0.085	0.172	0.479	1175.6	0.093	0.203	0.707	227.2
RA	0.117	0.140	0.542	1114.2	0.089	0.199	0.542	1160.4	0.093	0.205	0.709	226.5
MD	0.121	0.145	0.560	1106.1	0.092	0.207	0.561	1144.9	0.115	0.255	0.794	196.8
HC	0.031	0.022	0.852	49.6	0.001	0.001	0.913	1.52	0.015	0.040	0.977	1.8
BCN	0.124	0.138	0.699	913.7	0.090	0.177	0.702	909.4	0.114	0.254	0.889	136.5

for the Last.FM dataset. The diversity performances are improved even more significantly, especially for the personalisation $S(20)$ which has been improved for 43.72%, 54.37% and 26.12% for MovieLens, Netflix and Last.FM datasets respectively.

We further compare the recommendation performances of the optimised BCN index with that of classical similarity measures introduced in the Introduction section, including the CN, LHN, HP, AA, RA, MD, and HC indices. As shown in Table 3, the recommendation accuracies (precision $P(20)$ and recall $R(20)$) of the BCN index are comparable to the MD index which is normally considered as one of the most accurate algorithms. The BCN index is more accurate than many of the accuracy-based indices such as the CN, AA, RA. In terms of the diversity, the BCN index is comparable to the HC index, which is designed to achieve good diversity. While some diversity-based indices such as LHN, HP, and HC sacrifice the accuracy a lot to focus on the extreme unpopular recommendations, the proposed BCN index can achieve good accuracy and diversity simultaneously with reasonable preference on the popularity of recommended objects.

5 Discussion

Nowadays, countless valuable niche information is hidden in the dominance of popular information. While many channels are continuously enhancing the dominance of the popular information, such as the mass media and search engines [30,31], the recommender system is a chance to fulfil the service purpose to provide accurate recommendations for users, as well as enhance the accessibilities of niche information. However, the popularity bias of the similarity quantifications makes the recommendations also biased, leading to either poor accuracy or poor diversity.

Comparing a given bipartite network with the random network, this paper develops a BCN similarity index for the quantification of object similarities. The experiment results show that, the diversity of recommendations can be largely improved by balancing the popularity bias of the CN index. However, the accuracy will be sacrificed if removing all the random-driven common neighbours ($\lambda = 1$). To achieve good accuracy and diversity simultaneously, one should optimise the similarity quantification to remove the random-driven common neighbours in only

a certain degree. The optimised value λ_o for the MovieLens and Netflix are 0.33 and 0.36 respectively, which are less than the theoretical value $\lambda = 1$. On the other hand, the optimised value for the Last.FM is 1.4 which is larger than the theoretical value. Such difference between the Last.FM dataset with others may be raised from the different object degree distributions, and the extremely even user degree distribution ($H = 1$). Without hub users, objects are less likely to share many common neighbours, leading to the possibility that we may need a relatively larger value of λ to balance the popularity bias. With the optimised value, the accuracy and diversity of the recommendations can be simultaneously improved and are comparable to accuracy-based algorithms (such as MD) and diversity-based algorithms (such as HC) respectively.

This work is partially supported by a Key Project of National Natural Science Foundation of China (NSFC) with grant number 71532002.

Author contribution statement

LH and KL conceptualised the idea; LH conducted the theoretical calculation; LH and XP conducted the experiments; LH, XP and KL analysed the results and wrote the paper.

Open Access This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

References

1. D. Goldberg, D. Nichols, B.M. Oki, D. Terry, Commun. ACM **35**, 61 (1992)
2. P. Lops, M. De Gemmis, G. Semeraro, *Recommender systems handbook* (Springer, US, 2011)
3. L. Lv, M. Medo, C.H. Yeung, Y.C. Zhang, Z.K. Zhang, T. Zhou, Phys. Rep. **519**, 1 (2012)
4. G. Linden, B. Smith, J. York, IEEE Internet Comput. **7**, 76 (2003)

5. J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi et al., in *Proceedings of the 4th ACM Conference on Recommender Systems* (2010), pp. 293–296
6. F. CACHEDA, V. Carneiro, D. Fernández, V. Formoso, *ACM Trans. Web* **5**, 2 (2011)
7. K.H. Tso, L. Schmidt-Thieme, *Data science and classification* (Springer, Berlin/Heidelberg, 2006), pp. 271–278
8. M.J. Pazzani, D. Billsus, *The adaptive web* (Springer, Berlin/Heidelberg, 2007), pp. 325–341
9. C. Cattuto, V. Loreto, L. Pietronero, *Proc. Natl. Acad. Sci. USA* **104**, 1461 (2007)
10. H.N. Kim, A.T. Ji, I. Ha, G.S. Jo, *Electron. Commer. Res. Appl.* **9**, 73 (2010)
11. A. Zeng, A. Vidmer, M. Medo, Y.C. Zhang, *EPL* **105**, 58002 (2014)
12. L. Hou, K. Liu, J. Liu, R. Zhang, *Physica A* **468**, 415 (2017)
13. T. Zhou, J. Ren, M. Medo, Y.C. Zhang, *Phys. Rev. E* **76**, 046115 (2007)
14. J.G. Liu, L. Hou, X. Pan, Q. Guo, T. Zhou, *Sci. Rep.* **6**, 18653 (2016)
15. E.A. Leicht, P. Holme, M.E.J. Newman, *Phys. Rev. E* **73**, 026120 (2006)
16. E. Ravasz, A.L. Somera, D.A. Mongru, Z.N. Oltvai, A.L. Barabási, *Science* **297**, 1551 (2002)
17. L.A. Adamic, E. Adar, *Soc. Netw.* **25**, 211 (2003)
18. T. Zhou, L. Lü, Y.C. Zhang, *Eur. Phys. J. B* **71**, 623 (2009)
19. Y.C. Zhang, M. Blattner, Y.K. Yu, *Phys. Rev. Lett.* **99**, 154301 (2007)
20. A. Clauset, C. Moore, M.E.J. Newman, *Nature* **453**, 98 (2008)
21. H. Steck, in *Proceedings of the 5th ACM Conference on Recommender Systems* (2011), pp. 125–132
22. S.M. McNee, J. Riedl, J.A. Konstan, in *CHI'06 Extended Abstr. Hum. Factor Comput. Syst.* (ACM, 2006), pp. 1097–1101
23. X. Zhu, H. Tian, P. Zhang, Z. Hu, T. Zhou, *EPL* **111**, 48007 (2015)
24. J.G. Liu, T. Zhou, A. Guo, *Phys. Rev. E* **84**, 037101 (2011)
25. F. Chung, L. Lu, *Ann. Comb.* **6**, 125 (2002)
26. F. Chung, L. Lu, *Proc. Natl. Acad. Sci. USA* **99**, 15879 (2002)
27. J.G. Liu, Z.L. Hu, Q. Guo, *Eur. Phys. J. B* **86**, 478 (2013)
28. R. Guimerá, M. Sales-Pardo, L.A.N. Amaral, *Phys. Rev. E* **76**, 036102 (2007)
29. A. Vespignani, *Nat. Phys.* **8**, 32 (2012)
30. S. Fortunato, A. Flammini, F. Menczer, A. Vespignani, *Proc. Natl. Acad. Sci. USA* **103**, 12684 (2006)
31. A. Cornière, G. Taylor, *RAND J. Econ.* **45**, 576 (2014)