Collaborative filtering based on multi-channel diffusion

Ming-Sheng Shang [a] Ci-Hang Jin [b] Tao Zhou [b]
Yi-Cheng Zhang [a,b]

[a]*Lab of Information Economy and Internet Research, University of Electronic Science and Technology, 610054 Chengdu, China*

[b]*Department of Physics, University of Fribourg, Chemin du Musée 3, CH-1700 Fribourg, Switzerland*

Abstract

In this paper, by applying a diffusion process, we propose a new index to quantify the similarity between two users in a user-object bipartite graph. To deal with the discrete ratings on objects, we use a multi-channel representation where each object is mapped to several channels with the number of channels being equal to the number of different ratings. Each channel represents a certain rating and a user having voted an object will be connected to the channel corresponding to the rating. Diffusion process taking place on such a user-channel bipartite graph gives a new similarity measure of user pairs, which is further demonstrated to be more accurate than the classical Pearson correlation coefficient under the standard collaborative filtering framework.

Key words: recommender systems, collaborative filtering, diffusion-based similarity, complex networks, infophysics.
PACS: 89.75.Hc, 87.23.Ge, 05.70.Ln

1 Introduction

With the rapid growth of the Internet [1] and the World-Wide-Web [2], a huge amount of data and resource is created and available for the public. This, however, may result in a dilemma problem. On the one hand, the unprecedented growth of available information has brought us into the world of many possibilities: people may choose from thousands of movies, millions of books, and billions of web pages; on the other hand, the amount of information is increasing more quickly than our personal processing abilities and therefore

evaluations of all alternatives are not feasible at all. In consequence, it is vital to automatically extract the hidden information and make personalized recommendations.

A lot of work has been done in this field. A landmark is the use of *search engine* [3,4]. However, a search engine could only find the relevant web pages according to the input keywords and returns the same results regardless of users' habits and tastes. Another landmark is the so-called *recommender system* [5], which is essentially an information filtering technique that attempts to find out objects likely to be interesting to the target users. Due to its significance for economy and society, the design of efficient recommendation algorithms has become a common focus for computer science, mathematics, marketing practices, management science and physics (see the review articles [6,7,8] and the references therein).

Various kinds of recommendation algorithms have been proposed, including the content-based analysis [9], the spectral methods [10,11], the heat conduction algorithm [12], the opinion diffusion algorithm [13], the network-based inference [14,15], the latent semantic model [16], the latent Dirichlet allocation [17], the iterative self-consistent refinement [18], and so on. Among them, collaborative filtering (CF) is one of the earliest and the most successful algorithms underlies recommender systems [19]. A latent assumption of CF approach is that, in a social network, those who agreed in the past tend to agree again in the future. The most commonly used algorithmic framework of CF consists of two steps: firstly to identify the neighborhood of each user by computing similarities between all pairs of users based on their historical preferences, and then to predict by integrating ratings of target user's neighbors.

Algorithms within this framework differ in the definition of similarity, the formulation of neighborhoods and the computation of predictions. The most crucial ingredient in determining the accuracy of CF is how to properly quantify the similarity between user pairs [20]. In the simplest case, a recommender system can be well described by a bipartite user-object network [14], where the relations between users and objects are binary: either presence or absence. For example, in *Amazon.com* users are connected with books they purchased [21], and in *audioscrobbler.com* listeners are connected with the music groups they collected [22]. Under this bipartite case, the cosine similarity [23] is the most widely used index to quantify the proximity of user tastes. Recently, some new similarity indices are proposed and shown to be more accurate than the cosine similarity, including the random-walk-based similarities [20,24], the diffusion-based similarities [25,26,27], the transferring similarity [28], and so on. In addition, Fouss *et al.* [20] have demonstrated that some classical similarity indices, such as the Katz index [29] and the matrix forest similarity [30], can give really good recommendations under the framework of CF. However,
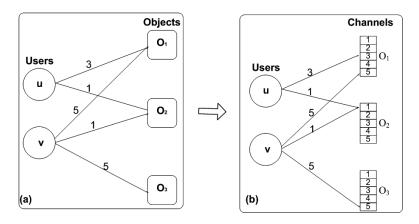
Fig. 1. Illustration of the two representations of a five-rating system. Plot (a) shows a routine representation where weights on edges denote the corresponding ratings. Plot (b) describes the multi-channel model where every object is divided into five channels, each of which represents a rating. User who votes an object is connected to the channel corresponding to the rating.

most of those indices are not easily to be exploited in measuring the user similarity of rating systems, where, instead of the simply binary correlations, users can vote objects by different ratings. For example, In *Yahoo music*, *Netflix.com* and *MovieLens*, people votes songs by discrete ratings from 1 to 5. In such rating systems, the *Pearson correlation coefficient* is the most widely used similarity measure [6]. In the calculation of the Pearson coefficient, each rating is treated as a number. Taking again the Yahoo Music as an example, the five ratings, from 1 to 5, corresponding to "Never play again", "It is ok", "I like it", "I love it", and "Can't get enough", and it is clear that the distance of feelings between "Never play again" and "It is ok" is much larger than the distance between "I love it" and "Can't get enough", however, when the ratings are treated as numbers, rich information gets lost and the distance between two neighboring ratings is supposed to be the same (in this example, it is one).

To best keep the original information, we divide every object into several channels, each of which represents a certain rating. Since most of the currently used recommendation engines adopt a five-rating system, this division will not bring much extra computational complexity. Figure 1 illustrates such a division for a five-rating system: Figure 1(a) is the routine representation with each object denoted by a node and the ratings are assigned to the corresponding edges, and Figure 1(b) is the new representation where each object is denoted by five channels corresponding to the five ratings. Under this representation, one can apply the diffusion process, usually only used in the bipartite version in the past [13], to the multi-channel systems. In this paper, to get the user similarity, we use a network-based resource-allocation method, which can be considered as a two-step diffusion process and thus much faster than the one based on a certain convergent condition [13]. We then use this user similarity

3

Table 1
Comparison of the two similarity indices on *MovieLens* and *Netflix*. The probe contains 10% of the total data, namely $p = 10$. All the number are obtained by averaging over five runs, each of which has an independently random division of training set and probe.

dataset	similarity index	RMSE	MAE
MovieLens	diffusion-based	0.9479	0.7415
	Pearson	1.0259	0.7805
NetFlix	diffusion-based	0.9406	0.7303
	Pearson	1.0441	0.7858

to predict ratings under the standard framework of collaborative filtering. We test this algorithm on two benchmark data sets, *MovieLens* and *Netflix*, the results demonstrate its advantage compared with the standard collaborative filtering adopting Pearson coefficient. This study indicates a strong potential of applying physical process to target one of the central scientific problems in the modern information science—how to automatically extract hidden information.

2   Method

In a recommender system, each user has voted some objects. Formally, let U be the set of m users, and O be the set of n objects, the rating of user $u \in U$ on object $\alpha \in O$ is denoted by $r_{u\alpha}$. We apply a resource-allocation process (two step of diffusion) to get similarities between users [14,31]. Given a user-channel bipartite network (see Fig. 1(b), such a network is consisted of m users and $5n$ channels), assuming that a certain amount of resource (e.g., recommendation power) is associated with each user, we will distribute this resource to other users via the channels. The process follows two steps. Firstly, each user distributes his initial resource evenly to all the channels he connects, and secondly, each channel distributes it's resource equally to all connected users.

Considering a bipartite graph $G = (U, C, E)$, where U is the set of users, C is the set of channels, and E is the set of edges connecting users and channels. After the first step, node $c \in C$ gets the fraction,

$$R_{cv} = \frac{a_{vc}}{k(v)}, \tag{1}$$

of resource from user v, where $k(v)$ is the degree of user v, $a_{vc} = 1$ if user v is
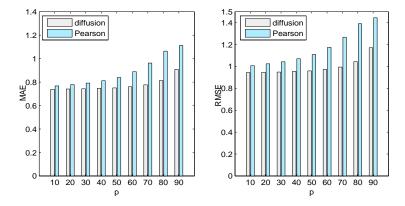
Fig. 2. Prediction accuracy on *MovieLens* for different densities of training set. All the number are obtained by averaging over five runs, each of which has an independently random division of training set and probe.

connected to channel c, and $a_{vc} = 0$ otherwise. Then, at the second step, each channel will distribute its resource to all the neighboring users. Thus, resource that user u gets from v, defined as the *similarity* between u and v, is:

$$s_{uv} = \sum_{c \in C} \frac{a_{uc} R_{cv}}{k(c)} = \frac{1}{k(v)} \sum_{c \in C} \frac{a_{uc} a_{vc}}{k(c)}, \tag{2}$$

where $k(c)$ is the degree of channel c. Note that, the similarity matrix $S = (s_{uv})$ is asymmetric, i.e., $s_{uv} \neq s_{vu}$. It is reasonable because a user who rated a lot of objects often has high probability to share many common channels with other users and thus will assign each of them lower weight. Actually, a recent empirical study showed that this kind of diffusion-based similarity can better describe the dependence between stations in the Chinese railway network, comparing with some traditional similarity measures [32]. In addition, the whole process obeys the conservation law, and the similarity matrix is column-normalized, as $\sum_u s_{uv} = 1$.

Once we have calculated the user similarities, we can then obtain the predicted rating on a new object $\alpha \in O$ for a target user $u \in U$ using the standard collaborative filtering framework, that is

$$r'_{u\alpha} = \bar{r}_u + \kappa \sum_v s_{uv}(r_{v\alpha} - \bar{r}_v), \tag{3}$$

where $\bar{r}_u$ denotes the average rating of user u, $\kappa = (\sum_v s_{uv})^{-1}$ serves as the normalization factor, and v runs over all users having voted the object α.
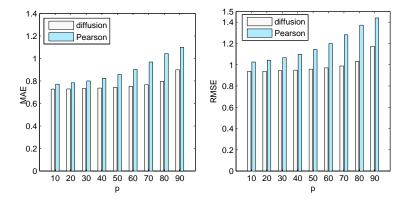
5

Fig. 3. Prediction accuracy on *Netflix* for different densities of training set. All the number are obtained by averaging over five runs, each of which has an independently random division of training set and probe.

3   Numerical results

To test the algorithmic accuracy, we use two benchmark data sets: (i)*MovieLens*, which consists of 943 users, 1682 objects, and 10^5 discrete ratings from 1 to 5. (ii)*Netflix*, which is a random sample of the original Netflix data set, containing 3000 users who have voted at least 45 objects, and 3000 movies having been voted at least by 23 users. There are in total $567,456$ ratings. We randomly divide this data set into two parts: one is the training set, treated as known information, and the other is the probe, whose information is not allowed to be used for prediction. we use a parameter, $p \in \{10, 20, .., 90\}$, to control the data density, that is, $p\%$ of the ratings are put into the probe set, and the remains compose the training set.

To evaluate the prediction accuracy, we use two well-known metrics: *mean absolute error* (MAE) and *root mean square error* (RMSE). They are respectively defined as:

$$MAE = \frac{1}{\|\mathcal{P}\|} \sum_{(u,\alpha)\in\mathcal{P}} (r_{u\alpha} - r'_{u\alpha}),$$

(4)

$$RMSE = \sqrt{\frac{1}{\|\mathcal{P}\|} \sum_{(u,\alpha)\in\mathcal{P}} (r_{u\alpha} - r'_{u\alpha})^2},$$

(5)

where $\mathcal{P}$ denotes the probe set.

We compare the proposed similarity with a benchmark one, namely the *Pearson correlation coefficient*, which has been proved highly competitive to other similarity methods and is widely used in collaborative filtering algorithms.

6

Under the Pearson's formula, the similarity, s_{uv}, between users u and v is

$$s_{uv} = \frac{\sum_\alpha (r_{u\alpha} - \bar{r}_u)(r_{v\alpha} - \bar{r}_v)}{\sqrt{\sum_\alpha (r_{u\alpha} - \bar{r}_u)^2}\sqrt{\sum_\alpha (r_{v\alpha} - \bar{r}_v)^2}}, \tag{6}$$

where α runs over all movies commonly voted by u and v.

Table 1 presents the algorithmic accuracies on *MoiveLens* and *Netflix*. Subject to the prediction accuracy, one can see that the diffusion-based similarity is notably better than the classical Pearson correlation coefficient for both data sets. Figure 2 and Figure 3 report the comparison between diffusion-based similarity and Pearson correlation coefficient for different data densities, namely different p. It can be seen that the diffusion-based similarity outperforms the Pearson correlation coefficient in all cases, and the difference becomes larger when the data gets sparser, indicating that this diffusion-based similarity has greater advantage for sparser systems.

4 Conclusion and Discussion

In this paper, by applying a diffusion process, we propose a new index to quantify the similarity between two users in a user-object bipartite graph. Under the standard collaborative filtering framework, we compare the diffusion-based similarity and the classical Pearson correlation coefficient. The numerical results on two benchmark data sets, *MovieLens* and *Netflix*, indicated that the diffusion-based similarity can better account for the proximity of user tastes and provide more accurate predictions. It is worthwhile to emphasize that the diffusion-based similarity can give competitively good predictions as the so-called *transferring similarity* based on Pearson correlation coefficient [28]. Since the transferring similarity, defined as $T = (I - \varepsilon S)^{-1}S$ with S the matrix of Pearson correlation coefficient and ε a free parameter, requires high computational resource and is parameter-dependent, the diffusion-based similarity, as a local and parameter-free index, is comparatively more efficient. We think the diffusion-based similarity, combined with the multi-channel representation, can find its application especially for the huge-size recommender systems with discrete ratings.

5 Acknowledgments

References

[1] G.-Q. Zhang, G.-Q. Zhang, Q.-F. Yang, S.-Q. Cheng, T. Zhou, New J. Phys. **10**, 123027 (2008).

[2] A. Broder, R. Kumar, F. Moghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Comput. Netw. **33**, 309 (2000).

[3] S. Brin, L. Page, Comput. Netw. ISDN Syst. **30**, 107 (1998).

[4] J. M. Kleinberg, J. ACM **46**, 604 (1999).

[5] P. Resnick, H. R. Varian, Commun. ACM **40**(3), 56 (1997).

[6] G. Adomavicius, A. Tuzhilin, IEEE Trans. Knowl. & Data Eng. **17**, 734 (2005).

[7] J. L. Herlocker, J. A. Konstan, K. Terveen, J. T. Riedl, ACM Trans. Inform. Syst. **22**, 5 (2004).

[8] J.-L. Liu, M. Z. Q. Chen, J. Chen, F. Deng, H.-T. Zhang, Z.-K. Zhang, T. Zhou, Int. J. Inform. &. Syst. Sci. **5**, 230 (2009).

[9] M. J. Pazzani, D. Billsus, Lect. Notes Comput. Sci. **4321**, 325 (2007).

[10] S. Maslov, Y.-C. Zhang, Phys. Rev. Lett. **87**, 248701 (2001).

[11] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Inf. Retr. **4**, 133 (2001).

[12] Y.-C. Zhang, M. Blattner, Y.-K. Yu, Phys. Rev. Lett. **99**, 154301 (2007).

[13] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, F. Yang, Europhys. Lett. **80**, 68003 (2007).

[14] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, Phys. Rev. E **76**, 046115 (2007).

[15] T. Zhou, L.-L. Jiang, R.-Q. Su, Y.-C. Zhang, Europhys. Lett. **81**, 58004 (2008).

[16] T. Hofmann, ACM Trans. Inform. Syst. **22**, 89 (2004).

[17] D. M. Blei, A. Y. Ng, M. I. Jordan, J. Mech. Learn. Res. **3**, 993 (2003).

[18] J. Ren, T. Zhou, Y.-C. Zhang, Europhys. Lett. **82**, 58007 (2008).

[19] J. B. Schafer, D. Frankowski, J. L. Herlocker, S. Sen, Lect. Notes Comput. Sci. **4321**, 291 (2007).

[20] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, IEEE Trans. Knowl. & Data. Eng. **19**, 355 (2007).

[21] G. Linden, B. Smith, J. York, IEEE Internet Comput. **7**, 76 (2003).

[22] R. Lambiotte, M. Ausloos, Phys. Rev. E **72**, 066107 (2005).

[23] G. Salton, M. J. McGill, *Introduction to Modern Information Retrieval* (MuGraw-Hill, Auckland, 1983).

[24] F. Gobel, A. Jagers, Stochastic Processes and Their Applications **2**, 311 (1974).

[25] Z. Huang, H. Chen, D. Zeng, ACM Trans. Inform. Syst. **22**, 116 (2004).

[26] R.-R. Liu, C.-X. Jia, T. Zhou, D. Sun, B.-H. Wang, Physica A **388**, 462 (2009).

[27] J.-G. Liu, B.-H. Wang, Q. Guo, Int. J. Mod. Phys. C **20**, 285 (2009).

[28] D. Sun, T. Zhou, R.-R. Liu, C.-X. Jia, J.-G. Liu, B.-H. Wang, arXiv: 0807.4495.

[29] L. Katz, Psychmetrika **18**, 39 (1953).

[30] P. Chebotarev, E. Shamis, Automation and Remote Control **58**, 1505 (1997).

[31] Q. Ou, Y.-D. Jin, T. Zhou, B.-H. Wang, B.-Q. Yin, Phys. Rev. E **75**, 021102 (2007).

[32] Y.-L. Wang, T. Zhou, J.-J. Shi, J. Wang, D.-R. He, Physica A **388**, 2949 (2009).