Published in "Europhysics Letters 88: 68008, 2009"
which should be cited to refer to this work.

Relevance is more significant than correlation: Information filtering on sparse data

Ming-Sheng Shang[1], Linyuan Lü[2], Wei Zeng[1,3], Yi-Cheng Zhang[2,3(a)] and Tao Zhou[2,4]

[1] *Web Sciences Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China - 610054 Chengdu, China*
[2] *Department of Physics, University of Fribourg - Chemin du Musée 3, CH-1700 Fribourg, Switzerland*
[3] *Lab of Information Economy and Internet Research, University of Electronic Science and Technology of China 610054 Chengdu, China*
[4] *Department of Modern Physics, University of Science and Technology of China - Hefei 230026, China*

PACS 89.20.Ff – Computer science and technology
PACS 89.75.Hc – Networks and genealogical trees
PACS 89.75.-k – Complex systems

Abstract – In some recommender systems where users can vote objects by ratings, the similarity between users can be quantified by a benchmark index, namely the *Pearson correlation coefficient*, which reflects the rating correlations. Another alternative way is to calculate the similarity based solely on the relevance information, namely whether a user has voted an object. The former one uses more information than the latter, and is intuitively expected to give more accurate rating predictions under the standard collaborative filtering framework. However, according to the extensive experimental analysis, this letter reports the opposite results that the latter method, making use of only the relevance information, can outperform the former method, especially when the data set is sparse. Our finding challenges the routine knowledge on information filtering, and suggests some alternatives to address the sparsity problem.

The explosion of information raises a serious overload problem: we face too many data and resources and are unable to efficiently find the relevant results. A promising way to solve this problem is to adopt *recommender systems* [1], which are essentially information filtering techniques that attempt 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, management science and physics (see the review articles [2] and the references therein). Many recommendation algorithms have been proposed, such as the content-based methods [3], spectral analysis [4,5], principle component analysis [6], iterative self-consistent refinement [7], heat conduction [8], opinion diffusion [9], network-based inference [10–12], and so on. A commonly existent problem for all recommendation algorithms, called the *sparsity problem*, is how to get accurate predictions for very sparse data [13].

Collaborative filtering (CF) is one of the most successful algorithms [2], whose basic assumption is that people who agreed in the past tend to agree again in the future [14]. Accordingly, the most important thing is to properly quantify the similarity between users. In some recommender systems, users are allowed to evaluate the objects by ratings. We call them *rating systems*. Obviously, the rating correlation can be considered as the similarity of users' tastes, that is, two users usually give close ratings to the same objects are considered to be similar. A standard index for this purpose is the so-called *Pearson correlation coefficient* (PCC). In comparison, the user similarity can be measured based only on the information whether a user has voted an object. The former one is called the *correlation-based index* while the latter is the *relevance-based index*. A big difference between these two indices lies in the fact that if two users give much different ratings to an object, it gives a negative contribution to the correlation-based index while a positive contribution to the relevance-based index.

Intuitively, the correlation-based index should give better predictions since it utilizes more information.

(a) E-mail: yi-cheng.zhang@unifr.ch

This routine thought is well accepted without any doubt as indicated by the previously proposed algorithms for rating systems [2] —most of the algorithms use PCC or its variants as standard similarity indices. We here argue that the relevance information may be more important than the rating correlations. The reasons are twofold: i) whatever the ratings are, to vote the same objects indicates a kind of taste similarity between users; ii) the ratings are very noisy while the relevance information is more credible (a bad mood may lead to a biased rating but not an inclination to read or vote a book with no interest). This letter reports extensive experimental tests on three data sets, *MovieLens*, *Netflix* and *Amazon*. Results are unexpected, that is, the relevance-based similarity index gives more accurate predictions than the PCC, especially when the data set is sparse.

A rating system can be represented by a bipartite network $G(U,O,E)$, where U, O and E are the sets of users, objects and links (labeled by ratings), respectively. We denote $r_{u\alpha}$ the rating from user u on object α. Let O_u be the set of objects that user u has voted and U_α the set of users having voted object α. The mean rating for u is $\bar{r}_u = \frac{1}{|O_u|}\sum_{\alpha\in O_u} r_{u\alpha}$. According to the standard collaborative filtering, the predicted rating of user u on an unvoted object α is

$$r'_{u\alpha} = \bar{r}_u + \kappa \sum_{v\in U_\alpha} s_{uv}(r_{v\alpha} - \bar{r}_v), \tag{1}$$

where s_{uv} denotes the similarity between user u and user v, and $\kappa = (\sum_v s_{uv})^{-1}$ is for normalization. The benchmark correlation-based index, PCC, is

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

where α runs over $O_u \cap O_v$.

For comparison, we here adopt a simple relevance-based similarity index. We first project the bipartite network into a monopartite network where two users are connected if they have voted at least one common object. Note that there are many refined methods to project the bipartite networks into weighted monopartite networks [10], but here we consider only the unweighted version for simplicity. Based on the projected network, we apply the *random walk with restart* (RWR) algorithm [15,16] to calculate the similarity between users. Consider a random walker starting from node i, who will iteratively move to a random neighbor with probability c and return to node i with probability $1-c$. Denote s_{ij} the probability this random walker locates at node j in the steady state, then we have

$$\vec{s_i} = cP^T\vec{s_i} + (1-c)\vec{e_i}, \tag{3}$$

where $\vec{e_i}$ is an $n\times 1$ vector (n is the number of users) with the i-th element equal to 1 and others all equal to 0, and P^T is the transition matrix[1] with $P_{ij} = 1/k_i$ if i and j are connected, and $P_{ij} = 0$ otherwise (k_i is the degree of node i). The solution is straightforward, as

$$\vec{s_i} = (1-c)(I - cP^T)^{-1}\vec{e_i}. \tag{4}$$

The probability s_{ij} here is used as a relevance-based similarity index. Different from the PCC in eq. (2), the RWR-based index is asymmetrical.

The RWR process is closely related to the famous PageRank algorithm [21]. Actually, the PageRank algorithm mimics the RWR process with a random walker initially located in each web page. Denoting by g_i the mass of random walkers in page i, the set of equations for the PageRank algorithm are

$$g_i = c\sum_{j\neq i}\frac{g_j}{k_j} + (1-c), \tag{5}$$

where j runs over all the web pages that contain an outlink pointing to i and k_j is the out-degree of page j. Despite of the close relationship between RWR process and PageRank algorithm, one should be aware of their essential difference. For example, $\vec{s_i}$ defined in eq. (3) is a vector representing the similarity between i and other nodes, where the walker starts from the node i. In contrast, g_i defined in eq. (5) is a scalar that measures the attractiveness of webpage i. The PageRank coefficient g_i can be considered as a centrality measure [22] of webpage i while the vector $\vec{s_i}$ cannot be directly related to any centrality indices.

We test the two indices on three data sets: i) *MovieLens*[2] is a movie recommendation website, which uses users' ratings to generate personalized recommendations. ii) *Netflix*[3] is an online DVD and Blu-ray Disc rental service in the US. The data we used is a random sample that consists of 3000 users who have voted at least 45 movies and 3000 movies having been voted at least by 23 users. iii) *Amazon*[4] is a multinational electronic commerce company. The original data were collected from 28 July 2005 to 27 September 2005, and what we used here is also a random sample. Table 1 summarizes the basic statistics.

To test the algorithm's accuracy, the observed ratings (links), E, is randomly divided into two parts: the training set, E^T, is treated as known information, while the probe set, E^P, is used for testing and no information in this set is allowed to be used for prediction. Clearly, $E = E^T \cup E^P$ and $E^T \cap E^P = \emptyset$. In our experiment, for each user we

[1] It is easy to prove that $\pi_i = k_i/2M$ with M the number of edges is a unique stationary distribution satisfying that $P^T\vec{\pi} = \vec{\pi}$ if the network is connected (see, for example, ref. [17]). However, one should note that because of the presence of loops in real networks [18,19] and the bipartite nature [20], starting with a random initial distribution, the stationary distribution may never be achieved. For example, if all the walkers are initially located in user nodes, the stochastic process driven by P^T will never converge.

[2] `http://www.grouplens.org`.

[3] `http://www.netflix.com`.

[4] `http://www.amazon.com`.

Table 1: The basic statistics of the three data sets. U, O and E are the total numbers of users, objects and ratings, respectively. The *density* equals $\frac{E}{U\times O}$.

Data Set	U	O	E	Density
MovieLens	943	1682	100000	6.3%
Netflix	3000	3000	197248	2.2%
Amazon	3000	3500	128193	1.2%

randomly select p% of his/her ratings as the probe set, and the remaining $(100-p)$% constitute the training set. One can control the data density by tuning p, with larger p corresponding to sparser data. To quantify the accuracy of predictions, we apply two standard metrics, *mean absolute error* (MAE) and *root-mean-square error* (RMSE):

$$\mathrm{MAE}=\frac{1}{\|E^P\|}\sum_{(u,\alpha)\in E^P}|r_{u\alpha}-r'_{u\alpha}|, \tag{6}$$

$$\mathrm{RMSE}=\sqrt{\frac{1}{\|E^P\|}\sum_{(u,\alpha)\in E^P}(r_{u\alpha}-r'_{u\alpha})^2}, \tag{7}$$

where $r_{u\alpha}$ is the real rating in the probe set, $r'_{u\alpha}$ is the predicted rating obtained by eq. (1), and $\|E^P\|$ is the number of user-object pairs in the probe set. Clearly, lower MAE and RMSE correspond to higher prediction accuracy.

Figure 1 presents the experimental results, wherein the larger p makes the training set sparser. For MovieLens and Netflix, the relevance-based index performs better when p exceeds a certain value p_c ($p_c\approx 40$ for MovieLens and $p_c\approx 15$ for Netflix), namely it performs better for sparser data. Since the density of Netflix is smaller than that of the MovieLens, the value of p_c for Netflix is also smaller. For the very sparse Amazon data, the relevance-based index always outperforms the correlation-based index in the monitored interval $10\leqslant p\leqslant 90$ (see footnote [5]). All these results indicate a surprising conclusion that the relevance-based index can outperform the correlation-based index, especially for the very sparse data. The practical significance of this finding is twofold. Firstly, to calculate the relevance-based indices is generally much faster than to calculate the correlation-based indices [15,16], and thus using the relevance-based indices instead of correlation-based indices can save time. Secondly, the real data are usually very sparse (usually sparser than the three data sets used in this letter, see, for example, the empirical analysis in ref. [23]), where the relevance-based indices

[5] When $p=90$, the training set is extremely sparse, with density equal to 0.12%. Two users rarely have chance to vote common objects, and thus the similarity matrices obtained by eq. (2) and eq. (4) are very sparse, leading to the predictions close to the simple average scores. This is the reason for the observed drop on PCC in $p=90$ for Amazon.

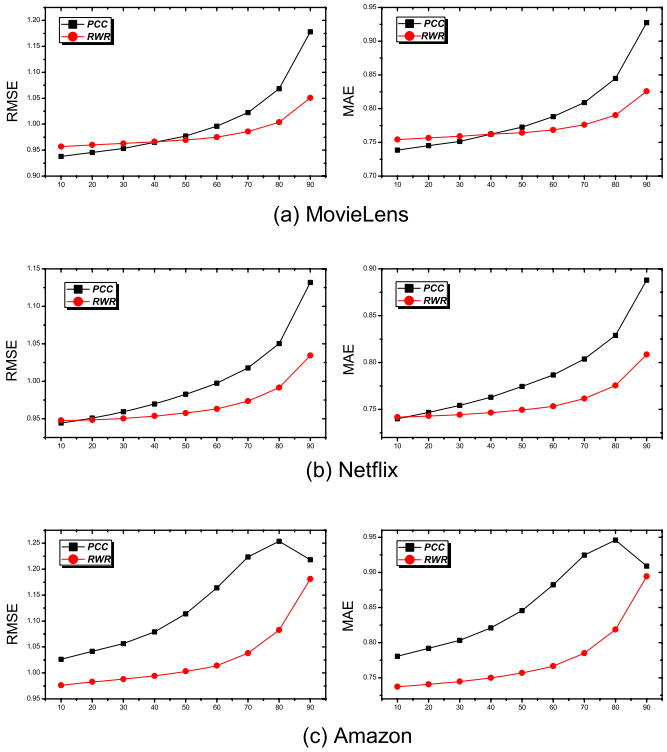

Fig. 1: (Colour on-line) Prediction accuracy as a function of p, where p% of the data constitute the probe set and thus a larger p corresponds to a sparser training data. Each point is obtained by averaging over 20 implementations with independently random divisions of training set and probe set. The parameter c in RWR is set as 0.95.

may get better predictions. Some refined correlation-based indices, like to assign a small positive contribution (instead of a negative contribution) to the similarity of two users if they vote a common object but with far different ratings, may lead to more accurate predictions. Analogously, considering properly weighted monopartite network may improve the prediction accuracy of the current correlation-based index. However, our motivation is not proposing a better similarity index for more accurate predictions, but aiming at highlighting the significance of the relevance information, even in the rating systems where the correlation information is available. This finding suggests some alternatives to address the sparsity problem in information filtering.

* * *

This work was partially supported by the Swiss National Science Foundation (200020-121848), and the National Natural Science Foundation of China (60973069, 90924011). M-SS acknowledges the China Postdoctoral Science Foundation (20080431273) and the Sino-Swiss Science and Technology Cooperation (SSSTC) Project (EG 20-032009). TZ acknowledges the National Natural Science Foundation of China (60744003, 10635040).

REFERENCES

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

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

[3] Pazzani M. J., *Artif. Intell. Rev.*, **13** (1999) 393.

[4] Sarwar B., Karypis G., Konstan J. A. and Riedl J. T., *Proceedings of the ACM WebKDD Workshop* (ACM Press, New York) 2000.

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

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

[7] Ren J., Zhou T. and Zhang Y.-C., *EPL*, **82** (2008) 58007.

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

[9] Zhang Y.-C., Medo M., Ren J., Zhou T., Li T. and Yang F., *EPL*, **80** (2007) 68003.

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

[11] Zhou T., Jiang L. L., Su R. Q. and Zhang Y.-C., *EPL*, **81** (2008) 58004.

[12] Zhou T., Su R.-Q., Liu R.-R., Jiang L.-L., Wang B.-H. and Zhang Y.-C., *New J. Phys.*, **11** (2009) 123008.

[13] Huang Z., Chen H. and Zeng D., *ACM Trans. Inf. Syst.*, **22** (2004) 116.

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

[15] Pan J.-Y., Yang H.-J., Faloutsos C. and Duygulu P., *SIGKDD'2004* (ACM Press, New York) 2004, pp. 653–658.

[16] Tong H., Faloutsos C. and Pan J.-Y., *ICDM'2006* (IEEE Press, Hong Kong) 2006, pp. 613–622.

[17] Lovász L., *Combinatorics*, **2** (1993) 1.

[18] Catanzaro M., Caldarelli G. and Pietronero L., *Phys. Rev. E*, **70** (2004) 037101.

[19] Bianconi G., Caldarelli G. and Capocci A., *Phys. Rev. E*, **70** (2004) 037101.

[20] Holme P., Liljeros F., Edling C. R. and Kim B. J., *Phys. Rev. E*, **68** (2003) 056107.

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

[22] Freeman L. C., *The Development of Social Network Analysis: A Study in the Sociology of Science* (Empirical Press, Vancouver Canada) 2004.

[23] Shang M.-S., Lü L., Zhang Y.-C. and Zhou T., arXiv: 0909.4938.