Personal Recommendation in User-Object Networks

Tao Zhou

Department of Physics, University of Fribourg, Chemin du Muse 3, CH-1700
Fribourg, Switzerland
Department of Modern Physics and Nonlinear Science Center,
University of Science and Technology of China, Hefei Anhui, 230026, P.R. China
zhutou@ustc.edu

Abstract. Thanks to the Internet and the World Wide Web, we live in a world of many possibilities we can choose from thousands of movies, millions of books, and billions of web pages. Far exceeding our personal processing capacity, this excessive freedom of choice calls for automated ways to find the relevant information. As a result, the field of information filtering is very active and rich with unanswered challenges. In this short paper, I will give a brief introduction on the design of recommender systems, which recommend objects to users based on the historical records of users' activities. A diffusion-based recommendation algorithm, as well as two improved algorithms are investigated. Numerical results on a benchmark data set have demonstrated the advantages in algorithmic accuracy.

Keywords: Infophysics, Personal Recommendation, Bipartite Networks, User-Object Networks, Diffusion.

1 Introduction

The last few years have witnessed an explosion of information that the exponential growth of the Internet and World Wide Web confronts us with an information overload: We face too much data and sources to be able to find out those most relevant for us. Indeed, we have to make choices from thousands of movies, millions of books, billions of web pages, and so on. Evaluating all these alternatives by ourselves is not feasible at all. As a consequence, an urgent problem is how to automatically find out the relevant objects for us, namely information filtering. A landmark for information filtering is the use of search engine, by which users could find the relevant web pages with the help of properly chosen keywords. However, the search engine has three essential disadvantages. First, it does not take into account personalization and returns the same results for people with far different habits. Therefore, if a user's habits are different from the mainstream, even with some *right keywords*, it is hard for him to find out what he likes from the countless searching results. Secondly, the search engine is a tool helping users to find out the web pages at least containing some content known to them. Many web pages, having potentialities to match a user's tastes,

J. Zhou (Ed.): Complex 2009, Part I, LNICST 4, pp. 247–253, 2009.

are, however, completely out of his horizon. In a word, the search engine is only helpful to find *what you know* instead of *what you like*, since you may have no idea of the latter. Thirdly, some tastes, such as the feelings of music and poem, can not be expressed by keywords, even language. The search engine, based on keyword matching, will lose its effectiveness in those cases.

To our knowledge, the most promising way to efficiently filter the overload information is to automatically provide personal recommendations based on the historical record of a user's activities [1,2]. For example, *Amazon.com* uses one's purchase record to recommend books, *AdaptiveInfo.com* uses one's reading history to recommend news, and *Recipefinder.com* uses one's stated interests to recommend restaurants. In a web-based serving system, a recommendation engine could improve loyalty by creating a value-added relationship between the site and the user. Actually, the more a user uses the recommendation engine – teaching it what he wants – the more loyal he is to the site. Recommendation engines also improve cross-sell for E-commerce systems by suggesting additional products for the customer to purchase. For example, the statistical investigation by *VentureBeat.com* shows that the recommendation engine in *Amazon.com* contributes about 35% of sales. Motivated by its significance in economy and society, the design of an efficient recommendation algorithm becomes a joint focus from engineering science to marketing practice, from mathematical analysis to physics community. Various kinds of recommendation algorithms have been proposed, including collaborative filtering [3], content-based analysis [4], spectral analysis [5], iteratively self-consistent refinement [6], principle component analysis [7], and so on.

In this short paper, I will introduce a diffusion-based algorithm for personal recommendation in bipartite user-object networks. Numerical results on a benchmark data set have demonstrated its advantage in algorithmic accuracy. Two improved algorithms are also introduced, which perform even better.

2 Diffusion-Based Algorithm

A recommendation system consists of users and objects, and each user has collected some objects. Denoting the object set as $O = \{o_1, o_2, \cdots, o_n\}$ and the user set as $U = \{u_1, u_2, \cdots, u_m\}$, the recommendation system can be fully described by a bipartite user-object network with $n + m$ nodes, where an object is connected with a user if and only if this object has been collected by this user. Connection between two users or two objects is not allowed. A Reasonable assumption is that the objects a user has collected are what he likes, and a recommendation algorithm aims at predicting his personal opinions (to what extent he likes or hate them) on those objects he has not yet collected. That is to say, given a target user, a recommendation algorithm should provide an ordered list of all the objects having not been collected by this user. Those objects in the top of this list are recommended to this user.

Based on the bipartite user-object network, an object-object network can be constructed, where each node represents an object, and two objects are connected if and only if they have been collected simultaneously by at least one user. We assume a certain amount of resource (i.e., recommendation power) is associated with each object, and the weight w_{ij} represents the proportion of the resource o_j would like to distribute to o_i. For example, in the book-selling system, the weight w_{ij} contributes to the strength of recommending the book o_i to a customer provided he has already bought the book o_j. The weight w_{ij} can be determined following a network-based diffusion process [8,9] where each object distributes its initial resource equally to all the users who have collected it, and then each user sends back what he has received equally to all the objects he has collected. For a general user-object network, the weighted projection onto the object-object network reads [9]:

$$w_{ij} = \frac{1}{k(o_j)} \sum_{l=1}^{m} \frac{a_{il} a_{jl}}{k(u_l)}, \tag{1}$$

where $k(o_j) = \sum_{i=1}^{n} a_{ji}$ and $k(u_l) = \sum_{i=1}^{m} a_{il}$ denote the degrees of object o_j and user u_l, and $\{a_{il}\}$ is the $n \times m$ adjacent matrix of the bipartite user-object network.

For a given user u_i, we assign some resource (i.e., recommendation power) on those objects already been collected by u_i. In the simplest case, the initial resource vector $\mathbf{f}$ can be set as

$$f_j = a_{ji}. \tag{2}$$

That is to say, if the object o_j has been collected by u_i, then its initial resource is unit, otherwise it is zero. After the resource-allocation process, the final resource vector is

$$\mathbf{f}' = W\mathbf{f}. \tag{3}$$

Accordingly, all u_i's uncollected objects o_j ($1 \leq j \leq n$, $a_{ji} = 0$) are sorted in the descending order of f'_j, and those objects with highest values of final resource are recommended.

To test the algorithmic accuracy, we use a benchmark data-set, namely *MovieLens*. The data consists of 1682 movies (objects) and 943 users, and users vote movies using discrete ratings 1-5. We therefore applied a coarse-graining method similar to that used in Ref. [10]: a movie has been collected by a user if and only if the giving rating is at least 3 (i.e. the user at least likes this movie). The original data contains 10^5 ratings, 85.25% of which are ≥ 3, thus after coarse gaining the data contains 85250 user-object pairs. To test the recommendation algorithms, the data set is randomly divided into two parts: The training set contains 90% of the data, and the remaining 10% of data constitutes the probe. The training set is treated as known information, while no information in the probe set is allowed to be used for prediction.

A recommendation algorithm should provide each user with an ordered queue of all its uncollected objects. For an arbitrary user u_i, if the relation $u_i - o_j$ is in the probe set (according to the training set, o_j is an uncollected object for u_i), we measure the position of o_j in the ordered queue. For example, if there are 1000 uncollected movies for u_i, and o_j is the 10th from the top, we say the position of o_j is 10/1000, denoted by $r_{ij} = 0.01$. Since the probe entries are actually collected by users, a good algorithm is expected to give high recommendations to them, thus leading to small r. Therefore, the mean value of the position value $\langle r \rangle$ (called *ranking score*, which approximately equals one minus the area under the receiver operating characteristic (ROC) curve [11]), averaged over all the entries in the probe, can be used to evaluate the algorithmic accuracy: the smaller the ranking score, the higher the algorithmic accuracy, and vice verse. The average values of ranking scores over 10 independent runs (one run here means an independently random division of data set) are 0.106, 0.122, and 0.140 for the present algorithm, the collaborative filtering[1], and the global ranking method[2], respectively. Clearly, the present diffusion-based algorithm performs the best.

3 Two Improved Algorithms

3.1 Diffusion-Based Algorithm with Tunable Initial Recommendation Power

Consider the initial resource located on object o_i as its assigned recommendation power. In the whole recommendation process, the total power given to o_i is $p_i = \sum_j f_i^j$, where the superscript j runs over all the users u_j. In the above mentioned algorithm, the total power of o_i is $p_i = \sum_j f_i^j = \sum_j a_{ij} = k(o_i)$. That is to say, the total recommendation power assigned to an object is proportional to its degree, thus the impact of high-degree objects (e.g., popular movies) is enhanced. Although it already has a good algorithmic accuracy, this uniform configuration may be oversimplified, and depressing the impact of high-degree objects in an appropriate way could, perhaps, further improve the accuracy. Motivated by this, we propose a more complicated distribution of initial resource to replace Eq. (2):

$$f_j^i = a_{ji} k^{\beta}(o_j), \tag{4}$$

where β is a tunable parameter. Compared with the original case, $\beta = 0$, a positive β strengthens the influence of large-degree objects, while a negative β weakens the influence of large-degree objects. In particular, the case $\beta = -1$ corresponds to an identical allocation of recommendation power ($p_i = 1$) for each object o_i.

[1] The collaborative filtering is based on measuring the similarity between users. For two users u_i and u_j, their similarity can be simply determined by $s_{ij} = \sum_{l=1}^{n} a_{li} a_{lj} / \texttt{min}\{k(u_i), k(u_j)\}$. For any user-object pair $u_i - o_j$, if u_i has not yet collected o_j (i.e., $a_{ji} = 0$), the predicted score, v_{ij} (to what extent u_i likes o_j), is given as $v_{ij} = \sum_{l=1,l\neq i}^{m} s_{li} a_{jl} / \sum_{l=1,l\neq i}^{m} s_{li}$. For any user u_i, all the nonzero v_{ij} with $a_{ji} = 0$ are sorted in descending order, and those objects in the top are recommended.

[2] The global ranking method sorts all the objects in the descending order of degree and recommends those with highest degrees.

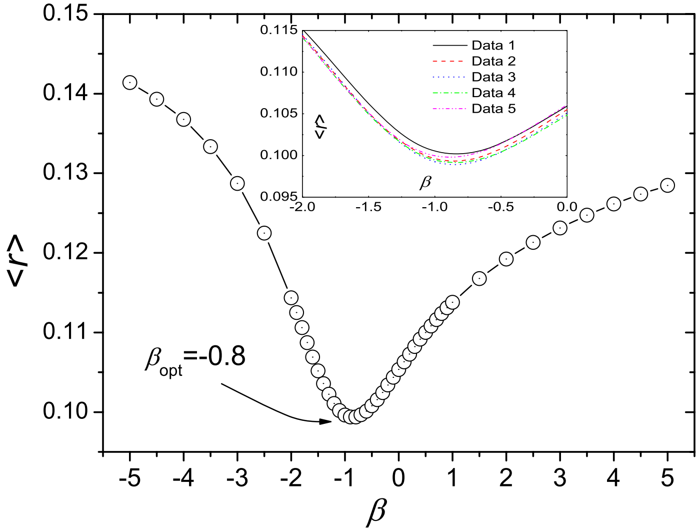

Fig. 1. (Color online) The ranking score $\langle r \rangle$ vs. β. The optimal β, corresponding to the minimal $\langle r \rangle \approx 0.098$, is $\beta_{\texttt{opt}} \approx -0.8$. All the data points shown in the main plot is obtained by averaging over five independent runs with different data-set divisions. The inset shows the numerical results of every separate run, where each curve represents one random division of data-set. After Ref. [12].

Ref. [12] reported the algorithmic accuracy as a function of β. As shown in Fig. 1, the curve has a clear minimum around $\beta = -0.8$. Compared with the uniform case, the ranking score can be further reduced by 9% at the optimal value. It is indeed a nice improvement for recommendation algorithms. Note that $\beta_{\texttt{opt}}$ is close to -1, which indicates that the more homogeneous distribution of recommendation power among objects may lead to a more accurate prediction.

3.2 Redundant-Eliminated Algorithm

In the diffusion-based algorithm mentioned in Section 2, for any user u_i, the recommendation value of an uncollected object o_j is contributed by all u_i's collected object, as

$$f'_j = \sum_l w_{jl} a_{li}. \tag{5}$$

Those contributions, $w_{jl}a_{li}$, may result from the similarities in same attributes, thus lead to heavy redundance. Generally speaking, if the correlation between o_i and o_k and the correlation between o_j and o_k contain some redundance to each other, then the two-step correlation between o_i and o_k, as well as that between o_j and o_k should be strong. Accordingly, subtracting the higher order correlations in an appropriate way could, perhaps, further improve the algorithmic accuracy. Motivated by this idea, we replace Eq. (5) by

$$\mathbf{f}' = (W + aW^2)\mathbf{f}, \tag{6}$$

where a is a free parameter. When $a = 0$, it degenerates to the algorithm in Section 2. If the present analysis is reasonable, the algorithm with a certain negative a could outperforms the case with $a = 0$.

Figure 2 reports the algorithmic accuracy, measured by the ranking score, as a function of a, which has a clear minimum around $a = -0.75$. Compared with the case in Section 2 (i.e., $a = 0$), the ranking score can be further reduced by 23% at the optimal value. This result strongly supports our analysis. It is worthwhile to emphasize that, 23% is indeed a great improvement for recommendation algorithms. The ultra accuracy of the present method, even far beyond our expectation, indicates a great significance in potential applications.

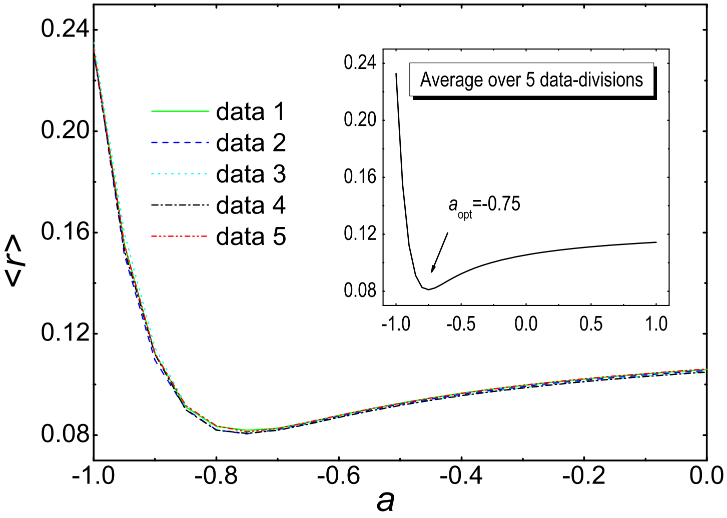

Fig. 2. The ranking score $\langle r \rangle$ *vs.* a. The main plot shows the numerical results of five independent runs, where each run corresponds to a random division of data set. The relation between $\langle r \rangle$ and a is very stable, and the fluctuation induced by the randomness in data division can be neglected. The curve shown in the inset is obtained by averaging over those five independent runs. The optimal a, corresponding to the minimal $\langle r \rangle \approx 0.0822$, is $a_{\mathtt{opt}} \approx -0.75$.

4 Conclusion

In this short paper, I introduced a diffusion-based personal recommendation algorithm, which performs obviously better than the commonly used collaborative filtering and global ranking method. In addition, I discussed two improved algorithms with remarkably higher accuracies. The former one has the same computation complexity as the original diffusion-based algorithm, while the latter one is ultra accurate. Those advantages are of significance in potentially real applications.

Acknowledgement

I acknowledge *GroupLens Research Group* for data (http://www.grouplens.org). This work is supported by the National Natural Science Foundation of China under Grant Nos. 60744003 and 10635040.

References

1. Konstan, J.A.: Introduction to recommender systems: Algorithms and Evaluation. ACM Trans. Inf. Syst. 22, 1–4 (2004)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 17, 734–749 (2005)
3. Herlocker, J.L., Konstan, J.A., Terveen, K., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Trans. Inform. Syst. 22, 5–53 (2004)
4. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. Lect. Notes Comput. Sci. 4321, 325–341 (2007)
5. Maslov, S., Zhang, Y.C.: Extracting Hidden Information from Knowledge Networks. Phys. Rev. Lett. 87, 248701 (2001)
6. Ren, J., Zhou, T., Zhang, Y.C.: Information Filtering via Self-Consisent Refinement. Europhys. Lett. 82, 58007 (2008)
7. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A Constant Time Collaborative Filtering Algorithm. Inf. Retr. 4, 133–151 (2001)
8. Ou, Q., Jin, Y.D., Zhou, T., Wang, B.H., Yin, B.Q.: Power-law strength-degree correlation from resource-allocation dynamics on weighted networks. Phys. Rev. E 75, 021102 (2007)
9. Zhou, T., Ren, J., Medo, M., Zhang, Y.C.: Bipartite network projection and personal recommendation. Phys. Rev. E 76, 046115 (2007)
10. Blattner, M., Zhang, Y.C., Maslov, S.: Exploring an opinion network for taste prediction: An empirical study. Physica A 373, 753–758 (2007)
11. Hanely, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36 (1982)
12. Zhou, T., Jiang, L.L., Su, R.Q., Zhang, Y.C.: Effect of initial configuration on network-based recommendation. Europhys. Lett. 81, 58004 (2008)