OPEN ACCESS Freely available online

PLOS ONE

Similarity from Multi-Dimensional Scaling: Solving the Accuracy and Diversity Dilemma in Information Filtering

Wei Zeng[1,2,3], An Zeng[3,5]*, Hao Liu[3], Ming-Sheng Shang[1,2]*, Yi-Cheng Zhang[1,3,4]

1 Web Sciences Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, **2** State Key Laboratory of Networking and Switching Technology, Beijing, P.R. China, **3** Department of Physics, University of Fribourg, Fribourg, Switzerland, **4** Institute of Information Economy, Hangzhou Normal University, Hangzhou, China, **5** School of Systems Science, Beijing Normal University, Beijing, P.R. China

Abstract

Recommender systems are designed to assist individual users to navigate through the rapidly growing amount of information. One of the most successful recommendation techniques is the collaborative filtering, which has been extensively investigated and has already found wide applications in e-commerce. One of challenges in this algorithm is how to accurately quantify the similarities of user pairs and item pairs. In this paper, we employ the multidimensional scaling (MDS) method to measure the similarities between nodes in user-item bipartite networks. The MDS method can extract the essential similarity information from the networks by smoothing out noise, which provides a graphical display of the structure of the networks. With the similarity measured from MDS, we find that the item-based collaborative filtering algorithm can outperform the diffusion-based recommendation algorithms. Moreover, we show that this method tends to recommend unpopular items and increase the global diversification of the networks in long term.

Citation: Zeng W, Zeng A, Liu H, Shang M-S, Zhang Y-C (2014) Similarity from Multi-Dimensional Scaling: Solving the Accuracy and Diversity Dilemma in Information Filtering. PLoS ONE 9(10): e111005. doi:10.1371/journal.pone.0111005

Editor: Jérémie Bourdon, Université de Nantes, France

Received May 7, 2014; **Accepted** September 19, 2014; **Published** October 24, 2014

Data Availability: The authors confirm that all data underlying the findings are fully available without restriction. The Movielens data is publicly available in www.Grouplens.com. The Netflix data is publicly available in www.netflix.com. The RYM data is publicly available in www.rateyourmusic.com.

Funding: This work is supported by the National Natural Science Foundation of China (Grant Nos. 61370150 and 91324002) and the Open Foundation of State Key Laboratory of Networking and Switching Technology (SKLNST-2013-1-18). W.Z. acknowledges support from the Program of Outstanding Ph.D. Candidate in Academic Research by UESTC (YBXSZC20131029). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* Email: an.zeng@unifr.ch (AZ); shang.mingsheng@gmail.com (MSS)

Introduction

Nowadays, individuals are confronted with a large amount of contents such that it is very time-consuming to find the needed information, which is known as the information overload problem. This problem becomes more serious as the rapid development of the Internet. To solve this problem, many information filtering techniques, such as search engines and recommender systems, are widely investigated. Specifically, recommender systems are a newly emergent technique which predicts what a user likes based on his/her historical choices.

Up to now, many recommendation algorithms have been proposed such as collaborative filtering (CF) [1–3], matrix factorization [4,5], spectral analysis [6], and so on. Some physical processes, including mass diffusion [7,8], heat conduction [9], were also introduced by physicists to design recommendation algorithms. A detailed summarization of recommender system technologies can be found in [10]. The most significant finding from these diffusion-based methods is that the hybridization of the mass diffusion and heat conduction can achieve both accurate and diverse recommendation [11]. This pioneer work was followed up later with many extensions such as the semi-local diffusion [12], the preferential diffusion [13], the biased heat conduction [14], network manipulation [15] and the item-oriented method [16]. Recently, the long-term influence of these diffusion-based recommendation methods on network evolution has also been studied [17,18].

Among the aforementioned algorithms, CF has been successfully applied in e-commerce [19,20]. The CF actually have two different versions: the user-based CF (UCF) and the item-based CF (ICF) [21–24]. The user-based CF estimates each user's preferences by referring to her similar users' tastes, while the item-based CF recommends items which are similar to the target user's selected items. Generally, the accuracy of the item-based CF is higher than that of the user-based CF. For both algorithms, the most important issue is how to qualify the similarities between users or items. There are many methods to measure the similarities of nodes based on network structure analysis including common neighbors, cosine index, Katz index, just to name a few [25,26]. However, these simple structural-based similarity measures are usually sensitive to the noisy information in networks, which results in a low recommendation accuracy. Moreover, some of these measures are strongly biased to large degree items, which makes the unpopular but relevant items be overlooked in the recommendation [3].

To solve the problems above, we make use of the multidimensional scaling (MDS) technique to estimate similarity between nodes. Online user-item bipartite networks are represented by a $M \times N$ adjacency matrix where M and N are respectively the number of users and items. Therefore, each item is described by a *M-dimensional* vector from the adjacency matrix. Based on MDS,

we design a method to map the *M-dimensional* item vectors into *H-dimensional* item vectors ($H \ll M$) and compute the similarities of item pairs in the *H-dimensional* space. There are two advantages: (1) The noise of data can be diminished by the dimension reduction, so that the similarity based on the low-dimensional space is more accurate than the high-dimensional space [27,28]. We compare the MDS method with the commonly-used cosine method in both artificial and real data, and find that the MDS method significantly outperforms the cosine method in estimating the item similarity. (2) MDS can remarkably speed up the computation of item similarity since we only have to deal with H-dimensional item vectors. Therefore, the MDS method can be used in the large-scale dataset. In fact, some other dimension reduction methods such as matrix factorization (MF) and singular value decomposition (SVD) have also been used in recommender systems [29,30]. In both methods, not only the item vectors but also the user vectors are considered. In most online systems such as user-movie rental systems, the number of users significantly exceeds the number of items. Therefore, it requires much more memory to store the user vectors than the item vectors. In other words, the MDS requires much less memory than the MF and SVD, making it more scalable.

We further apply the MDS to the item-based collaborative filtering algorithm. We test this method on real datasets and the results show that our method enjoys a considerably higher recommendation accuracy and diversity than the diffusion-based recommendation methods. Moreover, by investigating the network evolution driven by the recommendation algorithms, we found that our method could result in a more homogeneous item degree distribution in the long term.

Methods

Collaborate Filtering

A recommender system can be naturally described by a user-item bipartite network with the adjacency matrix $A_{M\times N}$ in which the element $a_{i\alpha}=1$ if the user i has collected the item α, and $a_{i\alpha}=0$ otherwise (To be consistent with previous papers, we use Greek and Latin letters, respectively, for item- and user-related indices) [2,31]. M and N are the number of users and items, respectively. The performance of ICF and UCF depends a lot on the similarity definition and the data sets [24,32]. We mainly focus on ICF in this paper, but parallel techniques can be applied in a user-oriented fashion.

The ICF provides each individual user with items which are similar to her selected items. That is, for user i, the recommendation score of item α is

$$p_{i\alpha}=\sum_{\beta=1}^{N} a_{i\beta}s_{\alpha\beta}, \tag{1}$$

where $s_{\alpha\beta}$ is the similarity between item α and β. Items will be sorted in descending order according to $p_{i\alpha}$ and the top-L items will be recommended to i. The most common way to compute $s_{\alpha\beta}$ is the cosine index [33,34], that is,

$$s_{\alpha\beta}=\frac{\mathbf{r}_\alpha\mathbf{r}_\beta^T}{\sqrt{(\mathbf{r}_\alpha\mathbf{r}_\alpha^T)(\mathbf{r}_\beta\mathbf{r}_\beta^T)}} \tag{2}$$

where $\mathbf{r}_\alpha$ and $\mathbf{r}_\beta$ are the α and β column of adjacency matrix A, respectively. This combination is referred as the ICF-cosine. There are some drawbacks of the standard cosine index and the ICF based on this index have some potential risks, which we will discuss later in the section.

Multi-dimensional Scaling

In bipartite networks, each item is characterized by the corresponding column of the adjacency matrix A, i.e. a M-dimensional vector. The goal of the MDS is to map the *M-dimension* vectors $A=\{\mathbf{r}_\alpha|\mathbf{r}_\alpha\in R^M,\alpha=1,2,...,N\}$ into the *H-dimension* vectors $Y=\{\mathbf{y}_\alpha|\mathbf{y}_\alpha\in R^H,\alpha=1,2,...,N\}$, such that dissimilarities from *M-dimension* space $d(\mathbf{r}_\alpha,\mathbf{r}_\beta)$ are well-approximated by the distances in the lower *H-dimensional* space $d(\mathbf{y}_\alpha,\mathbf{y}_\beta)$. The input of the MDS is an item $\times$ item dissimilarity (or similarity) matrix $D_{N\times N}=\{d(\mathbf{r}_\alpha,\mathbf{r}_\beta)\}$. One simple way to compute the $d(\mathbf{r}_\alpha,\mathbf{r}_\beta)$ is the Euclidean distance: $d(\mathbf{r}_\alpha,\mathbf{r}_\beta)=\|\mathbf{r}_\alpha-\mathbf{r}_\beta\|$. Given the dissimilarity matrix D, the task of the MDS is to minimize the cost function

$$E(Y)=\sum_{\alpha\beta}[d(\mathbf{y}_\alpha,\mathbf{y}_\beta)-d(\mathbf{r}_\alpha,\mathbf{r}_\beta)]^2, \tag{3}$$

where the $d(\mathbf{y}_\alpha,\mathbf{y}_\beta)=\|\mathbf{y}_\alpha-\mathbf{y}_\beta\|$ is the distance of item α and β from the *H-dimension* space. A well-known approach to find the solution is the Gradient Descent (GD) algorithm which repeatedly processes the iteration:

$$y_\alpha\leftarrow y_\alpha-\epsilon\nabla E_\alpha(Y), \tag{4}$$

Where

$$\nabla E_\alpha(Y)=2\sum_{\alpha\neq\beta}[d(\mathbf{y}_\alpha,\mathbf{y}_\beta)-d(\mathbf{r}_\alpha,\mathbf{r}_\beta)](\mathbf{y}_\alpha-\mathbf{y}_\beta)\frac{1}{d(\mathbf{y}_\alpha-\mathbf{y}_\beta)}, \tag{5}$$

and ∇ is the gradient operator. The step size ϵ should be small enough (e.g. 0.005).

Another kind of MDS takes into account the rank-order of the dissimilarities. That is, the Euclidean distances between points in Y approximate a monotonic transformation of the corresponding dissimilarities in D. Therefore, the cost function of this method is

$$E(Y)=\sum_{\alpha\beta}[\hat{d}(\mathbf{y}_\alpha,\mathbf{y}_\beta)-\hat{d}(\mathbf{r}_\alpha,\mathbf{r}_\beta)]^2, \tag{6}$$

where $\hat{d}(\mathbf{r}_\alpha,\mathbf{r}_\beta)$ is the monotonic transformation of $d(\mathbf{r}_\alpha,\mathbf{r}_\beta)$ using a least squares monotone regression algorithm called monotone fitting (MFIT), which is described in ref [35]. The MDS based on equation 3 is called *Metric MDS* (MMDS for short) and that based on equation 6 is called *Non-Metric MDS* (NMDS for short).

When recommending items to users, we apply the MDS (MMDS and NMDS) to measure the similarities of item pairs and then compute the recommendation score between user i and item α by equation 1. We refer this method as ICF-MDS. All i's uncollected items are sorted in descending order according to $p_{i\alpha}$ and the *top-L* items will be recommended to user i.

Diffusion-based Methods

The diffusion-based recommendation algorithms are commonly considered as the state-of-the-art approaches in both accuracy and diversity. The most representative one is the hybrid method (short for Hybrid) [11] which combines the mass diffusion (short for MD) [7] and heat conduction (short for HC) [9] processes. The hybrid method starts by assigning 1 unit resource to each selected item of

the target user, and 0 to the unselected items. Denoting the initial resource vector as $\vec{f}$, the resources will then diffuse in the user-item bipartite network according to $\vec{f}' = W\vec{f}$ where W is the diffusion matrix with each element

$$w_{\alpha\beta} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^{\lambda}} \sum_{i=1}^{M} \frac{a_{i\alpha} a_{i\beta}}{k_i}. \quad (7)$$

In above equation, λ is a tunable parameter. If $\lambda = 0$, it degenerates to the pure HC algorithm [9]. If $\lambda = 1$, it gives the MD algorithm [7]. The final resource vector $\vec{f}'$ will be sorted in the descending order and those items with most resources will be recommended.

In fact, the hybrid method is the same type of method as the cosine method. Given two items α and β, their cosine similarity is $s_{\alpha\beta} = \sum_{i=1}^{M} a_{i\alpha} a_{i\beta} / \sqrt{k_\alpha k_\beta}$. For the diffusion-based recommendation methods, the diffusion of resource on bipartite networks actually aims to calculate the similarity between items. Take the hybrid method as example, the resource that β receives from α reads $w_{\alpha\beta} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^{\lambda}} \sum_{i=1}^{M} \frac{a_{i\alpha} a_{i\beta}}{k_i}$, where k_α and k_i are the degree of item α and user i, respectively. λ is a tunable parameter. $w_{\alpha\beta}$ can be considered as the "similarity" between β and α. The cosine method is based on the calculation of the scalar product between two vectors. So the hybrid method can be regarded as a weighted scalar product between two vectors. Though there is an obvious commonness between these two methods, there is one important difference between them: the W matrix in the hybrid method is asymmetric while the S in the cosine method is symmetric. Different from the hybrid and cosine method, the MDS is based on Euclidean distance between two vectors. In principle, the distance between two vectors can be defined in other ways. We thus tried other distance definition in MDS, such as Euclidean Commute-Time Distance [36] and Hamming distance. We found that the Euclidean distance works best among these three (See table S3 in File S1).

Metric

The MovieLens data is used to test the algorithms' accuracy and diversity, which consists of 6040 users, 3900 movies and 1 million links (See table S1 in File S1). The results on other datasets are consistent with Movielens and presented in the supporting information material (See Fig. S1, Fig. S2 and table S4 in File S1). The data is randomly divided into two parts: the training set (E^T) and the probe set (E^P). The training set contains 80% of the original data and the recommendation algorithm runs on it. The rest of the data forms the probe set, which will be used to examine the recommendation performance. Measuring the accuracy and the diversity of top-L items in individuals recommendation list is actually more important from practical point of view since in real recommender systems individuals are only presented with top-L items. Accordingly, we employ four different metrics to measure accuracy and diversity of the top-L recommendation. A brief description of these four metrics is shown as follows:

Precision. For a target user i, the precision of recommendation, $P_i(L)$ is defined as $P_i(L) = d_i(L)/L$, where the $d_i(L)$ is the number of hit links, namely user i's associated links that are contained by both the probe set and the top-L recommendations. The precision of the whole system is the average of individual precisions over all users, given as $P(L) = \frac{1}{M} \sum_{i=1}^{M} P_i(L)$.

Recall. The recall of recommendation to i, $R_i(L)$, is defined as $R_i(L) = d_i(L)/E_i$, where E_i denotes the number of $u'_i s$ links in the probe set. Similarly, the recall of the whole system is defined as $R(L) = \frac{1}{M} \sum_{i=1}^{M} R_i(L)$. Higher precision and recall indicate higher accuracy of recommendations.

Hamming distance. This metric considers the uniqueness of different users' recommendation list. Given two users i and j, the difference between their recommendation lists can be $H_{ij}(L) = 1 - C_{ij}(L)/L$, where C_{ij} is the number of common items in the *top-L* places of both lists. Clearly, if user i and j have the same list, $H_{ij}(L) = 0$, while if their lists are completely different, $H_{ij}(L) = 1$. Averaging $H_{ij}(L)$ over all user pairs we obtain the mean distance $H(L)$, for which greater or lesser values mean, respectively, greater or lesser personalization of users' recommendation lists.

Novelty. This metric concerns the capacity of recommender systems to generate novel and unexpected results. Given an item α, its novelty is $I_\alpha = log(k_\alpha + 1)$. From this we can calculate the mean novelty $I_i(L)$ of each user's *top-L* items, and averaging over all users we obtain the mean novelty of the system $I(L)$.

Results

In this section, we will discuss the performance of MDS in estimating item similarities in both the artificial data and real data. For item α and β, one can get their similarity from the *M-dimension* space by equation 2. Their similarity from the *H-dimension* space can be obtained based on the Y computed by the MDS. That is,

$$s_{\alpha\beta} = \frac{\mathbf{y}_\alpha \mathbf{y}_\beta^T}{\sqrt{(\mathbf{y}_\alpha \mathbf{y}_\alpha^T)(\mathbf{y}_\beta \mathbf{y}_\beta^T)}}. \quad (8)$$

By comparing these two methods, one can identify which one performs better in quantifying the similarities between items. We normalized the similarities as follows:

$$\hat{s}_{\alpha\beta} = \frac{s_{\alpha\beta} - \min(s)}{\max(s) - \min(s)}, \quad (9)$$

where $\max(s)$ and $\min(s)$ are the maximum and minimum of all the similarities, respectively.

Simulations in Real Data

We carried out the simulations in an artificial data which consists of 500 users and 500 items. The results show that both MMDS and NMDS are significantly more accurate than the cosine method in estimating similarity between items (See Fig. S1 in File S1). We further compare the cosine and the MDS method on a real online bipartite network called MovieLens. The original data consists of 6040 users, 3900 movies and 1 million ratings. The rating matrix is transformed to *0–1* matrix where $r_{i\alpha} = 1$ if $r_{i\alpha} > 0$. We randomly select 500 movies and compute their similarities by the cosine, MMDS and NMDS methods, respectively. All the similarities are normalized by equation 9 and reported in Fig. 1. The movies are sorted according to their degrees in the ascending order. That's to say, the movies' degree increases from the left to the right in Fig. 1. For each movie, we then sort its similarities with other movies in the descending order, i.e., the value of similarity decreases from the top to the bottom in Fig. 1. The color denotes the value of similarity.

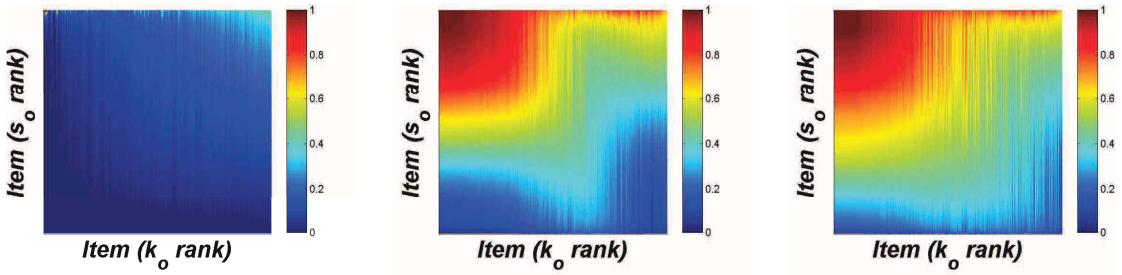

Figure 1. The compare of *cosine* and *MDS* (MMDS and NMDS) method in real data, MovieLens. All the movies are sorted by their degrees in a ascending order (horizontal ordinate). For a given movie α, other movies are sorted by their similarities with α in a ascending order (vertical ordinate) and the color depth denotes the value of similarity.
doi:10.1371/journal.pone.0111005.g001

One can see from the Fig. 1 that most similarities from the cosine method range from 0 to 0.2 and only a few of them are larger than 0.5, which indicates that the similarities between items are not well distinguished. The obtained item similarities based on the MMDS and NMDS share the same properties: Firstly, for each movie, its similarities with other movies vary significantly. Secondly, for those unpopular movies, their similarities with other movies tend to be very high and some of them are close to 1. But for those popular movies, their similarities with other movies are smaller. One possible reason is that the large degree movies have been collected by many users with different preferences. As a result, it is very difficult to identify which categories those movies belong to. Accordingly, their similarities with other movies are small.

Moreover, we present the relationship between average similarity $\langle S \rangle$ of an item to other items and its degree k_{item} in the top three figures in Fig. 2. It can be seen that in the cosine method $\langle S \rangle$ increases with k_{item}. In the MMDS method, $\langle S \rangle$ is roughly independent of k_{item}. In NMDS, $\langle S \rangle$ decreases with k_{item}. The distribution of similarity scores is also presented in the bottom three figures in Fig. 2. One can see that the similarity scores in the three methods are all homogeneously distributed. The mean of the distribution is around 0.5 in MMDS and NMDS, while the mean of the distribution is much smaller (around 0.1) in the cosine method.

Recommendation Accuracy and Diversity

We study the relationship between the accuracy and the dimension of Y computed by the MDS (See Fig. S2 in File S1). Our results show that the accuracy cannot be constantly increases by increasing H (when H is large enough, further enlarging H only includes noisy information). We also compared MDS to a matrix factorization method called the singular value decomposition (SVD) [37]. The SVD uses the k-largest singular values of A to construct a matrix A_k to approximate A. Here k is also the dimension of the obtained vectors from the decomposition. Normally, the optimal parameter k is determined by the number of largest singular values that are significantly larger than 0 [37]. After applying SVD to the movielens data, the results show that the singular value is close to zero when the dimension k exceeds 50. However, the best dimension number of the MDS method is around 100 (See Fig. S2 in File S1). The best dimension number obtained from SVD is different from that from MDS. This may due to the fact that the best dimension number in SVD and MDS (with ICF) is determined by different mechanisms: k in SVD is determined by the largest-singular values while H in MDS is determined by the recommendation precision.

We further compare our methods with the diffusion-based recommendation algorithms and the results are presented in the table 1. The accuracy of HC method is the lowest among these methods since it overwhelmingly focuses on the diversity of recommendation. The ICF-cosine is better than the MD but it is less effective than the Hybrid method. Among all the considered recommendation methods, the ICF-MMDS achieves the highest accuracy. More specifically, the ICF-MMDS method outperforms the ICF-cosine method by 19.7% and 27.9% in *precision*($L = 10$) and *recall*($L = 10$), respectively. These results confirm our previous conclusion that the similarity based on the MDS is better than the cosine index. We also carried out the simulation to compare MDS and cosine similarities under the UCF framework. Our experimental results show that UCF-MDS has higher recommendation accuracy than UCF. However, UCF-MDS is less effective than ICF-MDS (See table S2 in File S1).

In order to give more details about the ICF-MMDS and ICF-NMDS method, we study in detail the recommendation accuracy on users and items with different degrees. Since recall is defined based on users, it can be naturally used to measure the recommendation accuracy of the users with the same degree. When applied to items, we define the item recall as: $R_\alpha(L) = d_\alpha(L)/E_\alpha$ where E_α is the number of users who selected item α in the probe set, and $d_\alpha(L)$ is the number of times that α appears in these E_α users' recommendation lists. The recall of the items with the same degree is obtained by simply averaging $R_\alpha(L)$ of these items. The left figure of Fig. 3 gives the relationship between the accuracy and the movie degree. As one can seen, both ICF-MMDS and ICF-NMDS significantly improve the accuracy of small degree movies. Among all the methods, MD performs worst in recommending small degree movies. The right figure of Fig. 3 shows the relationship between the accuracy and the user degree. It can be seen that the ICF-MMDS and ICF-NMDS methods outperform others for both small and large degree users.

Our above results show that the ICF-MMDS and ICF-NMDS can improve the accuracy of those unpopular movies, which implies the recommendation from these two methods are diverse. The novelty and diversity results of those methods on MovieLens are presented in Fig. 4. The left figure gives the results of *Novelty*, where it can be seen that the best method with respect to *Novelty* is HC. On the contrary, the *Novelty* of the MD and ICF-cosine are not satisfactory enough. The *Novelty* of ICF-MMDS and ICF-NMDS increases with the dimension H, which indicates that they provide more novel movies with a smaller H. The right figure gives the recommendation diversity measured by the *Hamming Distance*. Different from the *Novelty*, the best method is the ICF-NMDS rather than the HC method. The diversity of both

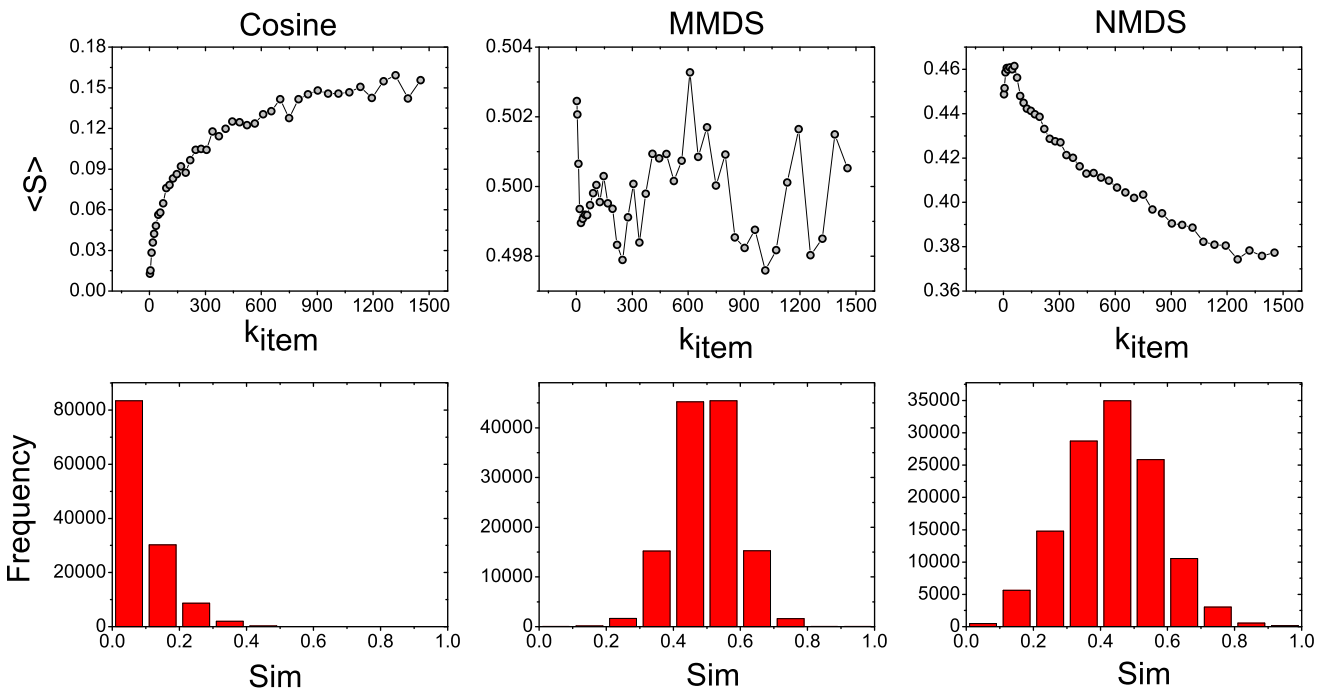

Figure 2. The relationship between average similarity of an item to other items and its degree, as well as the distributions of similarity scores under different methods.
doi:10.1371/journal.pone.0111005.g002

ICF-MMDS and ICF-NMDS methods decreases with the dimension H but still better than others when H is large.

Effect on Network Evolution

Moreover, we study impacts of recommendation algorithms on the long-term diversification of user-item bipartite network. We again randomly sample 500 movies from the MovieLens data. For each user, we provide her with top-10 ranked movies by the recommendation algorithm and assume that she will randomly select one of them. As a result, each user's degree will be increased by 1. We repeat this scenario for 10 times and then investigate the changes of each movie's degree distribution as well as the corresponding *Gini* coefficient. The left figure of Fig. 5 gives the changes of movies' degrees in the zipf plot. The *Origin* curve denotes movies' degrees in the original bipartite network. Other curves denote the movies' degrees after 10 times of the above recommendation processes. We observe that the top-100 popular movies' degrees are greatly increased by the MD and ICF-cosine algorithms while the degree increment of other movies is very small. It means the unpopular movies are overlooked while popular movies are mostly recommended by these two methods. The HC algorithm mainly increases the degrees of those unpopular movies, which is opposite to the algorithm of MD and ICF-cosine. Different from the previous methods, the degrees of both the popular and unpopular movies are increased by the ICF-MMDS and ICF-NMDS. Between ICF-MMDS and ICF-NMDS, one can see that the degree increment of unpopular movies by the ICF-NMDS is more than that by the ICF-MMDS, which indicates that the ICF-NMDS works better in recommending the fresh movies for users.

The changes of *Gini* coefficient of the system is presented in the right figure of Fig. 5. Suppose $\mathbf{k}$ is the movie degree vector sorted in the ascending order, the *Gini* coefficient of the system is

$$G = \frac{1}{n}(n+1-2\frac{\sum_{\alpha=1}^{n}(n+1-\alpha)k_{\alpha}}{\sum_{\alpha=1}^{n}k_{\alpha}}), \qquad (10)$$

where n is the size of $\mathbf{k}$. The *Step* in the figure denotes the number

Table 1. The accuracy compare results of different recommendation approaches on MovieLens.

Method	Precision($L=10$)	Precision($L=20$)	Recall($L=10$)	Recall($L=20$)
ICF-MMDS	**0.3507**	**0.2844**	**0.1604**	**0.2412**
ICF-NMDS	0.3338	0.2716	0.1506	0.2284
MD	0.2355	0.1900	0.1006	0.1528
HC	0.0024	0.0235	0.0014	0.0186
Hybrid	0.3256	0.2673	0.1492	0.2325
ICF-cosine	0.2929	0.2323	0.1254	0.1853

The recommendation length L is set to 10 and 20. The dimensions of both ICF-MMDS and ICF-NMDS are 100. The λ of Hybrid method is 0.2.
doi:10.1371/journal.pone.0111005.t001

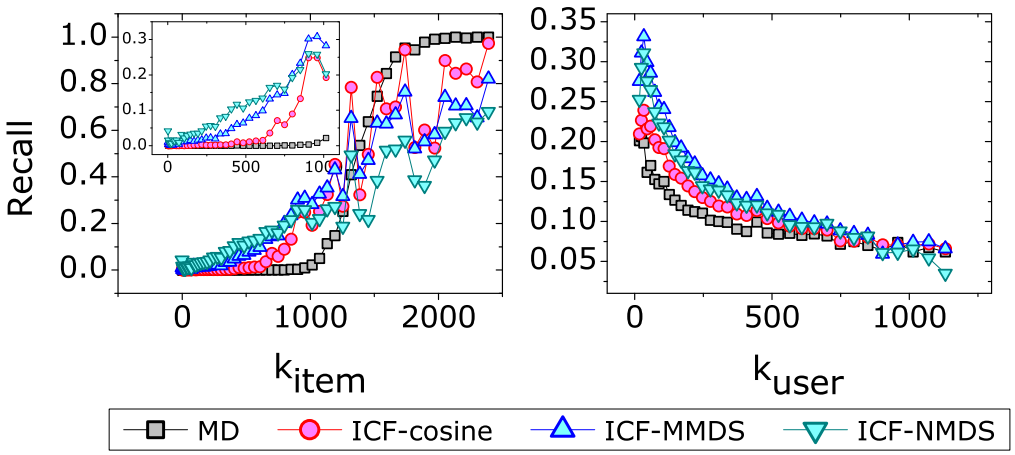

Figure 3. The relationship between accuracy and the user degree (k_{user}) and movie degree (k_{item}). For a given x, its corresponding *recall* is obtained by averaging all the users whose degrees are in the range of $[a(x^2-x),a(x^2+x)]$, where a is chosen as $\frac{1}{2}\log 5$. The recommendation length is 20 and the dimension of MMDS and NMDS is set to 30.
doi:10.1371/journal.pone.0111005.g003

of iterations of the above recommendation process. If $Step = 0$, the *Gini* coefficient is computed by the original movie degrees. For MD and ICF-cosine, the Gini coefficient grows fast after each recommendation process. This "rich gets richer" result in fact contradicts to the concept of personalized recommendation which is supposed to guide users' attention to different items according to their personal tastes. The HC algorithm decreases the *Gini* of the system in the long-term since it mainly recommends those unpopular movies to users. For both the ICF-MMDS and ICF-NMDS, the *Gini* coefficient stays relatively stable in long term.

Complexity of Recommendation Algorithms

We finally discuss the computational complexity of our methods. The complexity of computing the distance matrix is $O(M^2N^2)$ where M and N are the number of user and items, respectively. There are $N \times N$ entries in the distance matrix, therefore the complexity of computing the low-dimension matrix Y by the gradient descent method is $O(H^2N^2)$ where H is the dimension of Y. We test the methods on an i5-2500 dual-core processor 3.3 GHz PC. For the MovieLens data set, it spends 571.6 s in total to compute the Y by the MDS method and only 0.6041 s to calculate the similarities over all item pairs when $H = 100$. However, it takes 340.9 s to compute item similarities by the traditional cosine method. From the definition of mass diffusion and hybrid method, they have the same computational complexity with CF method as the resource diffusion process can be considered as the computation of item similarities. To obtain the transition matrix, it takes 319.8 s and 525.2 s for the mass diffusion and hybrid method, respectively. Although the total running time of the MDS-based method is more than the traditional methods, the computation of Y can be done off-line. When providing on-line recommendation service for users, we can use the pre-stored Y to calculate the item similarities and recommend items by CF method.

Additionally, we show in Fig. 6 the computation time of different methods when the network size is increased. Starting

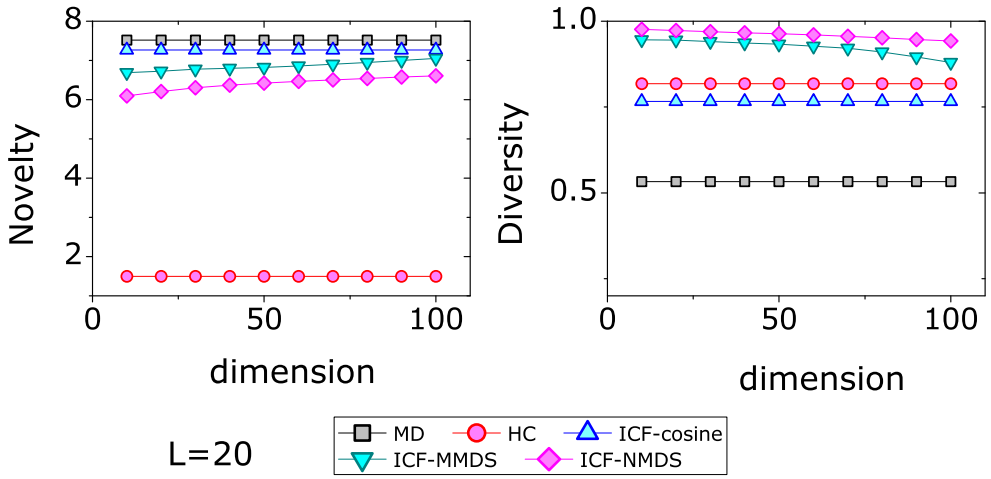

Figure 4. The diversity results of different recommendation approaches on MovieLens. The recommendation length is set to 20.
doi:10.1371/journal.pone.0111005.g004

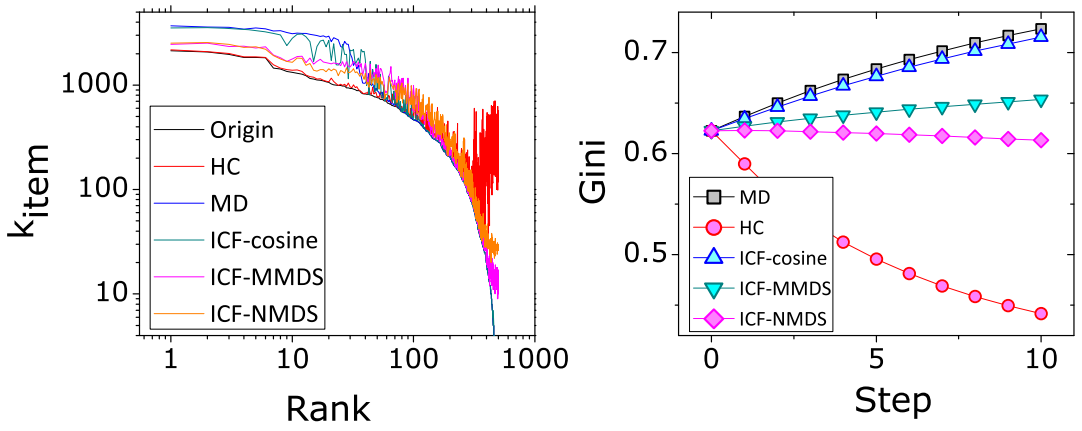

Figure 5. The changes of each movie's degree and the *Gini* index of the system. The dimension of MMDS and NMDS is set to 30.
doi:10.1371/journal.pone.0111005.g005

from the real data, we add some ratio of artificial users with degree equal to the mean degree of the existing users. The links of new users randomly connect to the items. Fig. 6 shows the relation between the computation time for the item similarity and the ratio of new users. From the figure, one can see that the computation time of traditional methods (cosine, diffuse and hybrid) increases with the number of new users in the system. Although the running time of MDS training process (computing Y matrix) is increased with the user number, the running time of computing the item similarity matrix is barely affected, as shown in the inset in Figure 6. As we discussed above, the computation of Y matrix can be done off-line and the computing the item similarity matrix is done online. Therefore, the recommendation speed of the ICF-MDS method is independent of user number in real application.

Discussion

The collaborative filtering method is considered as the most popular and already widely applied to e-commerce. The performance of CF strongly depends on the approach of computing the users' or items' similarity. In the literature, there are many handy similarity measures such as common neighbor index and its variants. However, theses methods cannot smooth out noise, which may result in a distorted estimation of the similarity between nodes. To solve this problem, we apply the multi-dimensional scaling method to measure similarity. The method first maps the items from high dimension to low dimension, then compute the item similarity from the low dimension space. This mapping process can effectively eliminate the noisy information from data and result in a more accurate recommendation when applied to item-based collaboration filtering method. Moreover, the computing complexity of similarity from the low-dimension space is much lower than that from the high-dimension space, which efficiently accelerates the speed of recommendation. Finally, we study the long term diversification of the resulted bipartite networks when different recommendation methods are repeatedly used. We find the ICF based on MDS can lead to a relatively stable degree distribution of the items, which may help to form a healthy information ecology in practice.

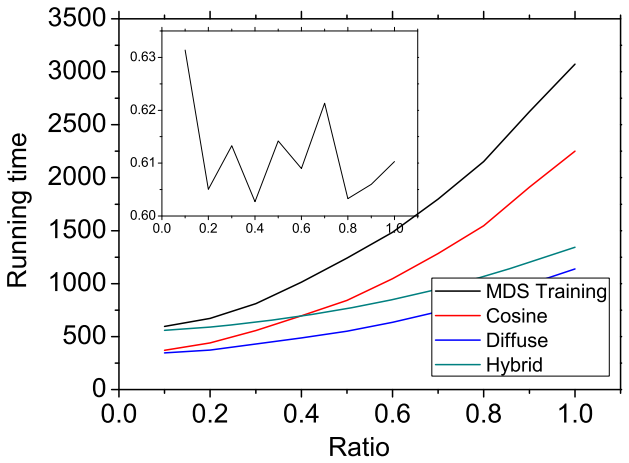

Figure 6. The computation time of methods with the increasing of network size. Inset gives the running time of computing the item similarity matrix by Y when the dimension number is 100.
doi:10.1371/journal.pone.0111005.g006

Supporting Information

File S1 Combined file of supporting figures and tables. **Figure S1.** The compare of *cosine* and *MDS* (MMDS and NMDS) method in artificial data. The dimension H of Y is increased from 2 to 50. **Figure S2.** The relationship between the accuracy (precision and recall) and the dimension H of Y computed by the MDS. The recommendation length is 20. **Figure S3.** The relationship between the accuracy (precision and recall) and the dimension of Y computed by the MDS on Netflix and RYM dataset. The recommendation length L is 20. **Figure S4.** The diversity and novelty results on the Netflix and RYM data. The recommendation length L is 20. **Table S1.** The MDS-based method under the UCF framework. The dimensions of ICF-MDS and UCF-MDS are 100 and 200, respectively. **Table S2.** MDS based on different distance computations.
(PDF)

Author Contributions

Conceived and designed the experiments: AZ MSS YCZ. Performed the experiments: WZ HL. Analyzed the data: WZ AZ. Wrote the paper: WZ AZ MSS YCZ.

References

1. Gediminas A, Alexander T (2005) 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.
2. Shang MS, Lü LY, Zeng W, Zhang YC, Zhou T (2009) Relevance is more significant than correlation: Information filtering on sparse data. EPL 88: 68008.
3. Zeng W, Shang MS, Zhang QM, Lü LY, Zhou T (2010) Can dissimilar users contribute to accuracy and diversity of personalized recommendation? Int J Mod Phys C 21: 1217–1227.
4. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42: 30–37.
5. Hu YF, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. ICDM '08, pp. 263–272.
6. Maslov S, Zhang YC (2001) Extracting hidden information from knowledge networks. Phys Rev Lett 87: 248701.
7. Zhou T, Ren J, Medo M, Zhang YC (2007) Bipartite network projection and personal recommendation. Phys Rev E 76: 046115.
8. Zhang YC, Medo M, Ren J, Zhou T, Li T, et al. (2007) Recommendation model based on opinion diffusion. EPL 80: 68003.
9. Zhang YC, Blattner M, Yu YK (2007) Heat conduction process on community networks as a recommendation model. Phys Rev Lett 99: 154301.
10. Lü LY, Medo M, Yeung CH, Zhang YC, Zhang ZK, et al. (2012) Recommender systems. Phys Rep 519: 1–49.
11. Zhou T, Kuscsik Z, Liu JG, Medo M, Wakeling JR, et al. (2010) Solving the apparent diversity-accuracy dilemma of recommender systems. Proc Natl Acad Sci USA 107: 4511–4515.
12. Zeng W, Zeng A, Sheng SM, Zhang YC (2013) Information filtering in sparse online systems: Recommendation via semi-local diffusion. PLoS ONE 8: e79354.
13. Lü LY, Liu WP (2011) Information filtering via preferential diffusion. Phys Rev E 83: 066119.
14. Liu JG, Zhou T, Guo Q (2011) Information filtering via biased heat conduction. Phys Rev E 84: 037101.
15. Zhang FG, Zeng A (2012) Improving information filtering via network manipulation. EPL 100: 58005.
16. Qiu T, Chen G, Zhang ZK, Zhou T (2011) An item-oriented recommendation algorithm on coldstart problem. EPL 95: 58003.
17. Zeng A, Yeung CH, Shang MS, Zhang YC (2012) The reinforcing influence of recommendations on global diversification. EPL 97: 18005.
18. Zhao DD, Zeng A, Shang MS, Gao J (2013) Long-term effects of recommendation on the evolution of online systems. Chin Phys Lett 30: 118901.
19. Schafer JB, Konstan J, Riedl J (1999) Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce. New York, NY, USA: ACM, EC '99, pp. 158–166.
20. Linden G, Smith B, York J (2003) Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput 7: 76–80.
21. Wang J, de Vries AP, Reinders MJT (2006) Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: ACM, SIGIR '06, pp. 501–508.
22. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22: 5–53.
23. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., UAI'98, pp. 43–52.
24. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web. New York, NY, USA: ACM, WWW '01, pp. 285–295.
25. Lü L, Jin CH, Zhou T (2009) Similarity index based on local paths for link prediction of complex networks. Phys Rev E 80: 046122.
26. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Am Soc Inf Sci Technol 58: 1019–1031.
27. Kinoshita K, Obayashi T (2009) Multi-dimensional correlations for gene coexpression and application to the large-scale data of arabidopsis. Bioinformatics 25: 2677–2684.
28. Bcavin C, Tchitchek N, Mintsa-Eya C, Lesne A, Benecke A (2011) Improving the efficiency of multidimensional scaling in the analysis of high-dimensional data using singular value decomposition. Bioinformatics 27: 1413–1421.
29. Cremonesi P, Koren Y, Turrin R (2010) Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems. New York, NY, USA: ACM, RecSys '10, pp. 39–46.
30. Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender Systems. New York, NY, USA: ACM, RecSys '10, pp. 135–142.
31. Zhou T, Medo M, Cimini G, Zhang ZK, Zhang YC (2011) Emergence of scale-free leadership structure in social recommender systems. PLoS ONE 6: e20648.
32. Koren Y (2008) Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '08, pp. 426–434.
33. Lü L, Zhou T (2011) Link prediction in complex networks: A survey. Physica A 390: 11501170.
34. Zhou T LL, Zhang YC (2009) Predicting missing links via local information. Eur Phys J B 71: 623–630.
35. Kruskal J (1964) Nonmetric multidimensional scaling: A numerical method. Psychometrika 29: 115–129.
36. Fouss F, Pirotte A, Renders JM, Saerens M (2007) Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans on Knowl and Data Eng 19: 355–369.
37. Berry MW, Dumais ST, O'Brien GW (1995) Using linear algebra for intelligent information retrieval. SIAM Rev 37: 573–595.