Mach Learn (2008) 72: 231–245
DOI 10.1007/s10994-008-5068-4

A collaborative filtering framework based on both local user similarity and global user similarity

Heng Luo · Changyong Niu · Ruimin Shen · Carsten Ullrich

Received: 22 June 2008 / Revised: 22 June 2008 / Accepted: 23 June 2008 / Published online: 8 July 2008

Abstract Collaborative filtering as a classical method of information retrieval has been widely used in helping people to deal with information overload. In this paper, we introduce the concept of local user similarity and global user similarity, based on surprisal-based vector similarity and the application of the concept of maximin distance in graph theory. Surprisal-based vector similarity expresses the relationship between any two users based on the quantities of information (called *surprisal*) contained in their ratings. Global user similarity defines two users being similar if they can be connected through their locally similar neighbors. Based on both of Local User Similarity and Global User Similarity, we develop a collaborative filtering framework called LS&GS. An empirical study using the MovieLens dataset shows that our proposed framework outperforms other state-of-the-art collaborative filtering algorithms.

Keywords Collaborative filtering · Similarity measure · Information theory

1 Introduction

Collaborative filtering algorithms are widely applied on e-commerce web sites, where they predict user preferences of items taking into consideration the opinions (in the form

---

Editors: Walter Daelemans, Bart Goethals, Katharina Morik.

H. Luo (✉) · C. Niu · R. Shen · C. Ullrich
Department of Computer Science and Technology, Shanghai Jiao Tong University, 1954 Huashan Road, Shanghai 200030, China
e-mail: hengluo@sjtu.edu.cn

C. Niu
e-mail: cyniu@sjtu.edu.cn

R. Shen
e-mail: rmshen@sjtu.edu.cn

C. Ullrich
e-mail: ullrich_c@sjtu.edu.cn

of preference ratings) of other "similar" users. Generally, there are two major classes of collaborative filtering algorithms, memory-based algorithms and model-based algorithms (Breese et al. 1998). Because of their simplicity and robustness, memory-based algorithms are widely applied in practice, e.g. (Herlocker et al. 1999; Linden et al. 2003; Resnick et al. 1994). To estimate a prediction for a particular user (i.e., an active user), the memory-based algorithms first find users from the database that are most similar to this active user, and then combine those ratings together. The measurement techniques of the similarity between users include the Pearson Correlation Coefficient (Resnick et al. 1994), Vector Space Similarity (VSS) algorithm (Breese et al. 1998), and the extended generalized vector space model (Soboroff and Nicholas 2000). These algorithms can be considered as *user-based* algorithms.

However in practice, systems based on collaborative filtering algorithms often face the problem of having at their disposal only an insufficient amount of preferences ratings of their individual users. Therefore, one of the biggest challenges of designing a collaborative filtering system is how to provide accurate recommendations with the sparse user profile data. To estimate an active user's rating of a particular item, traditional *user-based* methods first find the user's neighbors (the users who are similar to the active user). Then, the active user's rating is predicted by averaging the (weighted) known ratings on the item by his/her neighbors. This kind of methods is based on the assumption that similar users have similar rating patterns. Unfortunately, due to the data sparsity problem, firstly, often there does neither exist a sufficient amount of similar neighbors, nor a sufficient amount of ratings of the particular item.

The measurement of the similarity between users plays a fundamental role in *user-based* algorithms (Resnick et al. 1994; Wang et al. 2006; Jin et al. 2004). Traditional methods of computing similarity, however, have two important shortcomings. Firstly, usually all items are treated the same when computing the similarity of users. This is addressed by (Jin et al. 2004), which assign different weights to items in order to allow for items to contribute in different strength to the user similarity calculation. The second problem is that the similarity of two users cannot be calculated if they have not rated any identical item. In other words, due to the data sparsity problem, the neighbors of active user cannot be found. To solve this problem, it seems promising to transitively examine whether the neighbors of the two users are similar. That means we should estimate similarities between any two users from a global perspective.

In this paper, we address these two problems by proposing to divide user similarity into two parts, namely local user similarity and global user similarity. Local similarity is determined based on surprisal-based vector similarity (SVS). In SVS, the rating of each item is firstly modeled as a Laplacian random variable. Then the quantities of information (surprisal) contained in the ratings of a specific user will be used to represents his/her preference. The similarity of any two users' surprisal vector is defined as the local similarity of them. We will show that some of the ratings of the same item carry more discriminative information than others. Furthermore, we argue that less common ratings for a specific item tend to provide more discriminative information than the most common ratings. Second, the global similarity measures the similarity of two users by further considering the extent to which their neighbors are locally similar (using the local similarity). In this way, the global similarity takes the data sparsity problem in consideration by propagating similarity measurement. All local similarities of any two users represented as the weights of edges will be used to construct a user graph. The global similarity can be calculated as the maximin distance of any two nodes in the graph. Under global user similarity, two users become more similar if they can be connected through a series of locally similar neighbors. In brief, the local similarity attempts to accurately measure the similarity of two users' preference. The global similarity tries to find more similar users when the data of user's preference is sparse. On this basis, we propose a collaborative filtering framework that employs both Local User Similarity and Global User Similarity (LS&GS).

The major contributions of this paper are as follows.

(1) We propose a novel method (SVS) to compute local user similarity.
(2) We apply Maximin distance to capture global relationships of users to address the problem of data sparsity.
(3) A collaborative filtering framework (LS&GS) is proposed to based on local user similarity and global user similarity.

The remainder of our paper is organized as follows: Sect. 2 will introduce the necessary background and related work. In Sect. 3, we will present the definition of the local user similarity and the global user similarity. Section 4 introduces the proposed collaborative filtering framework. In Sect. 5, the experimental results are provided, followed by the conclusions in Sect. 6.

2 Notations and related work

There are two major classes of collaborative filtering algorithms: memory-based and model-based approaches (Breese et al. 1998). Memory-based algorithms make recommendations based on the entire user profile database. Model-based algorithms, in contrast, use a compact model which usually was previously learned from the user profile database to produce recommendations.

In this section, we describe the most relevant existing approaches of memory-based algorithms and briefly introduce the model-based algorithms. First, we describe the notations that are used throughout this paper.

Given a recommendation system consisting of M users and N items, there is a $M \times N$ user-item matrix R. Each entry $r_{m,n} = x$ represents the rating that user m gives to item n, where $x \in \{1, 2, \ldots, r_{\max}\}$. The default $r_{m,n}$ value, meaning that the rating is unknown, is 0.

The user-item matrix can be decomposed into row vectors:

$$R = [u_1, \ldots, u_M]^T, \qquad u_m = [r_{m,1}, \ldots, r_{m,N}]^T, \quad m = 1, \ldots, M.$$

The row vector u_m represents the ratings of user m for all of N items.

Alternatively, the matrix can also be represented by its column vectors:

$$R = [i_1, \ldots, i_N]^T, \qquad i_n = [r_{1,n}, \ldots, r_{M,n}]^T, \quad n = 1, \ldots, N.$$

The column vector i_n represents the ratings of item m by all of M users.

2.1 Memory-based approaches

Memory-based algorithms were applied successfully in various real-life applications (Herlocker et al. 1999; Linden et al. 2003). The major types of memory-based approaches are *user-based* approaches (Breese et al. 1998) and *item-based* approaches (Linden et al. 2003; Sarwar et al. 2001). The former approaches form a heuristic implementation of the "Word of Mouth" phenomenon (Shardanand and Maes 1995). The later one attempts to improve the scalability of collaborative filtering algorithms. *User-based* collaborative filtering predicts an active user's interest in a particular item based on rating information from similar user profiles (Breese et al. 1998; Herlocker et al. 1999; Resnick et al. 1994). Each user profile corresponds to a row vector sorted in the user-item matrix. In detail, *user-based* approaches first calculate all similarities of any two row vectors. For predicting a user's rating of a particular item, a set of top-N similar users can be identified. Those top-N users' ratings for the item will be averaged as the prediction by weighted.

Consequently, the predicted rating $\hat{r}_{a,y}$ of test item y by test user a is computed as

$$\hat{r}_{a,y} = \frac{\sum_{k=1}^{K} w_{a,u_k} r_{u_k,y}}{\sum_{k=1}^{K} |w_{a,u_k}|}$$

where w_{a,u_k} denotes the similarity between the test user and his neighbors u_k.

Item-based approaches use the similarity between items instead of users. First, the similarity of items (column vectors in the user-item matrix) can be calculated. Then the unknown ratings can be predicted by averaging the ratings of other similar items rated by this active user. That is

$$\hat{r}_{a,y} = \frac{\sum_{k=1}^{K} w_{y,i_k} r_{a,i_k}}{\sum_{k=1}^{K} |w_{y,i_k}|},$$

where w_{y,i_k} indicates the similarity between the test item and the most similar items i_k.

Similarity computation methods, such as the Pearson Correlation Coefficient (PCC) algorithm (Resnick et al. 1994) and the Vector Space Similarity (VS) algorithm (Breese et al. 1998) are applied in *user-based* and *item-based* methods.

The PCC method defines the similarity between two users w_{u_p,u_q} as

$$w_{u_p,u_q} = \frac{\sum_{\{i|r_{p,i},r_{q,i}\neq 0\}} (r_{p,i} - \bar{r}_p)(r_{q,i} - \bar{r}_q)}{\sqrt{\sum_{\{i|r_{p,i},r_{q,i}\neq 0\}} (r_{p,i} - \bar{r}_p)^2} \cdot \sqrt{\sum_{\{i|r_{p,i},r_{q,i}\neq 0\}} (r_{q,i} - \bar{r}_q)^2}},$$

where $\bar{r}_p$ denotes the mean of user p's ratings.

While the VS method defines the similarity as

$$w_{u_p,u_q} = \frac{\sum_{\{i|r_{p,i},r_{q,i}\neq 0\}} r_{p,i} r_{q,i}}{\sqrt{\sum_{\{i|r_{p,i}\neq 0\}} r_{p,i}^2} \cdot \sqrt{\sum_{\{i|r_{q,i}\neq 0\}} r_{q,i}^2}}.$$

2.2 Model-based approaches

The model-based algorithms present good scalability once they have built the model. However, the overhead introduced for building and updating the model should be counted in when evaluating this kind of algorithms. Various popular model-based algorithms exist, such as the aspect model (AM) (Hofmann and Puzicha 1999), the Personality Diagnosis model (PD) (Pennock et al. 2000) and the User Rating Profile model (URP) (Marlin 2004a).

The aspect model (Hofmann and Puzicha 1999) is a probabilistic latent-space model, which models individual preferences to a convex combination of preference factors. The latent class variable is associated with each observation pair of a user and an item. The aspect model assumes that users and items are independent from each other given the latent class variable. However, the aspect model cannot perform inference on novel user profiles (Marlin 2004b). In other words, in order to make predictions for novel users, AM has to be retrained based on the new training set, which should include the ratings of novel users.

The Personality diagnosis approach (Pennock et al. 2000) considers each user in the user-item matrix as an individual model. To predict the unknown rating of an item by an active user, PD first calculates the likelihood for the active user to be in the 'model' of each training user and then uses the aggregate average of ratings for the item by the training users as the estimator.

The User Rating Profile model (Marlin 2004a) is a generative, latent variable model which represents each user as a mixture of user attitudes, and the mixing proportions are distributed according to a Dirichlet random variable. URP is different from AM by making novel users' rating predication possible.

3 Local user similarity and global user similarity

The key to many memory-based approaches is to estimate the similarity between two users (Resnick et al. 1994; Jin et al. 2004). In this section, we will first introduce our method called surprisal-based vector space similarity to compute local user similarity. Then, addressing the data sparsity problem, global user similarity will be proposed. Global user similarity makes two users to become more similar if they can be connected through their locally similar neighbors.

3.1 Local user similarity (surprisal-based vector space similarity)

The Pearson Correlation Coefficient (PCC) algorithm is widely applied in collaborative filtering algorithms to compute user similarity (Breese et al. 1998; Linden et al. 2003; Resnick et al. 1994; Wang et al. 2006; Sarwar et al. 2001).

Breese et al. (1998) proposed that items with similar ratings should have less important impact in determining user similarity than those with different ratings. They suggested using the Inverse User Frequency as the weights of items. Herlocker et al. (1999) adopted variance weighting to improve PCC. The results turned out be to slightly worse than with no weighting (Herlocker et al. 1999).

The ratings of a specific item are usually centralized around an average attitude. In the PCC algorithm, if two users give an item the same rating, these two ratings will make the two users more similar. We argue that we need to additionally consider the difference between the rating and the average attitude. If the rating is close to the average attitude, the rating only represents that these two users act like most other people. Based on the rating we cannot conclude that the preferences of these two users are similar or dissimilar. On the other hand, if the rating is totally different from the average attitude, the rating will provide more discriminative information to determine whether their preferences are similar. Intuitively, a rarely given rating for an item will be extremely useful to help us distinguish the user which gives the unexpected rating from other users. For example, the movie "Godfather" is highly favored by lots of people. The fact that a user likes the movie tells us almost nothing about his/her preference. In contrast, if a user dislikes the movie and gives it a very low rating (i.e., the kind of rating that is rare) for it, we can easily distinguish him/her from others and know something about his/her preference (e.g., that he/she maybe dislike mafia movies).

Although most users' ratings of a specific item are centralized around an average attitude, there still exist some users who give much higher (or lower) ratings than the average attitude. In other words, the distribution of the ratings has fat tails. To implement the intuition above, we modeled the rating of each item as Laplacian random variables $Laplace(\bar{u}_i, b_i)$ rather than Gaussian random variables. The probability density function of the Laplacian random variable is

$$f(r|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|r-\mu|}{b}\right) = \frac{1}{2b}\begin{cases}\exp(-\frac{\mu-r}{b}) & \text{if } r < \mu,\\ \exp(-\frac{r-\mu}{b}) & \text{if } r > \mu.\end{cases}$$

Here, μ is a location parameter and $b > 0$ is a scale parameter. Given M ratings, independent and identically distributed samples $r_{1,i}, r_{2,i}, \dots, r_{M,i}$, then using the maximum likelihood estimator, estimators of μ_i and b_i are expressed as (Norton 1984)

$$\hat{\mu}_i = \frac{1}{M}\sum_{p=1}^{M} r_{p,i},$$

$$\hat{b}_i = \frac{1}{M}\sum_{p=1}^{M} \left|r_{p,i} - \hat{\mu}_i\right|.$$

We propose a method for computing local user similarity based on the users' surprisal vector, rather than on the users' ratings vector. User p's surprisal vector S_p is defined as following

$$\begin{aligned}S_p &= [s_{p,1}, \dots, s_{p,N}]^T\\ &= [\operatorname{sgn}(r_{p,1} - \hat{\mu}_1) * I(r_{p,1}), \dots, \operatorname{sgn}(r_{p,N} - \hat{\mu}_N) * I(r_{p,N})]^T, \quad p = 1, \dots, M\end{aligned}$$

where $\operatorname{sgn}(r_{p,1} - \hat{\mu}_i)$ presents whether the attitude of user p about item i is positive or negative in comparison with the average attitude about the item, and $I(r_{p,i})$ is the quantity of information (surprisal) of the rating $r_{p,i}.I(r_{p,i})$ is defined as

$$I(r_{p,i}) = -\ln(f(r = r_{p,i}|\hat{\mu}_i, \hat{b}_i)) = \ln(2\hat{b}_i) + \frac{|r_{p,i} - \hat{\mu}_i|}{\hat{b}_i}.$$

Given the users' surprisal vectors, we can adopt the Vector SPACE Similarity (VS) algorithm to calculate the user local similarity. We call this method surprisal-based vector similarity (SVS), which is defined as

$$sim_L(u_p, u_q) = \frac{\sum_{\{i|r_{p,i}, r_{q,i} \neq 0\}} s_{p,i} * s_{q,i}}{\sqrt{\sum_{\{i|r_{p,i}, r_{q,i} \neq 0\}} s_{p,i}^2} \cdot \sqrt{\sum_{\{i|r_{p,i}, r_{q,i} \neq 0\}} s_{q,i}^2}}.$$

Ma et al. (2007) proposed to add a correlation significance weighting factor that would devalue similarity weights that were based on a small number of co-rated items,

$$sim'_L(u_p, u_q) = \frac{Min(|I_{u_p} \cap I_{u_q}|, \gamma)}{\gamma} sim_L(u_p, u_q)$$

where $|I_{u_p} \cap I_{u_q}|$ is the number of items which user u_p and user u_q rated in common. If the number of co-rated items is smaller thanγ, the similarity of these users will be devalued. This change avoids overestimating the similarities of users who have rated a few items identically, but may not have similar overall preferences.

The method is adopted to compute the local user similarity called surprisal-based vector similarity with significance weighting (SVSS).

In this paper, we aim to emphasize that less common ratings for a specific item tend to provide more discriminative information than the most common ones. With regard to the choice of the distribution for modeling ratings, some sophisticated variations of the Laplacian distribution are available (Kotz et al. 2001).

3.2 Global user similarity

Under this similarity, we can find more neighbors of an active user even when he/she has few immediate neighbors using local user similarity. To attain this, we first construct a user graph using the local similarity as the weight of edges. Then, we use the maximin distance of two users in the graph as the measurement of the global similarity between them.

3.2.1 User graph

We construct a user graph that describes their relationships, as follows.

Definition 1 (User graph) A user graph is an undirected weighted graph $G = (U, E)$, where

(1) U is the node set (each user is regarded as a node of the graph G);
(2) E is the edge set. Associated with each edge $e_{pq} \in E$, w_{pq} is a weight subject to $w_{pq} > 0$, $w_{pq} = w_{qp}$.

In this paper, we employ local user similarity as the weights of edges,

$$w_{pq} = \begin{cases} sim'_L(u_p, u_q) & \text{if } sim'_L(u_p, u_q) > 0, \\ 0 & \text{else.} \end{cases}$$

3.2.2 Maximin distance on user graph

Given a user graph $G = (U, E)$, a path from node u_p to $u_q (u_p, u_q \in U)$ is a sequence of links, $P_{pq} = (u_p, \dots, u_i, \dots, u_q), u_p, u_i, u_q \in U$. If there are K paths between nodes u_p and u_q, these paths will be indicated as $P^1_{pq}, P^2_{pq}, \dots, P^K_{pq}$. Given a path between u_p and u_q the minimal hop distance of these nodes along any path P^j_{pq} is defined as follow:

$$minimalhop_j(u_p, u_q) = \min_{u_i, u_{i+1} \subset P^j_{pq}} w_{i,i+1}, \quad \forall u_i, u_{i+1} \in P^j_{pq}, 1 \le j \le k.$$

The maximal value of the two nodes' minimal hop distance along any paths is called the maximin distance of the two nodes,

$$\begin{aligned} maximinhop(u_p, u_q) &= \max_{k=1,\dots,K} minimalhop_k(u_p, u_q) \\ &= \max_{k=1,\dots,K} \left\{ \min_{u_i, u_{i+1} \subset P^k_{ij}} w_{i,i+1} \right\}, \quad \forall u_i, u_{i+1} \in P^k_{pq}. \end{aligned}$$

The corresponding path is called as maximin path.

The global similarity of two users is defined as the maximin distance between them:

$$sim_G(u_p, u_q) = maximinhop(u_p, u_q).$$

For any two users u_p and u_q, if $sim_G(u_p, u_q) \neq 0$, it means there are d users forming a sequence $S = \{(u_p, u_1), \ldots, (u_{d-1}, u_k), (u_d, u_q)\}$, and $\forall (u_i, u_j) \in S, sim_L(u_i, u_j) \geq sim_G(u_p, u_q)$. It can be interpreted as meaning that user u_p finds a similar user u_q through $u_1, \ldots, u_k$, while all of these are similar in sequence. From this we can derive the following propositions.

Proposition 1 $\forall (u_p, u_q) \in U, sim_G(u_p, u_q) \geq 0$

Proof $\forall u_p, u_q \in U, w_{pq} \geq 0$

$$sim_G(u_p, u_q) = \max_{k=1,\ldots,K} \left\{ \min_{u_i, u_{i+1} \subset P_{ij}^k} w_{i,i+1} \right\}, \quad \forall u_i, u_{i+1} \in P_{pq}^k,$$

$$sim_G(u_p, u_q) \geq 0.$$

□

Proposition 2 $\forall u_p, u_q \in U, sim_G(u_p, u_q) \geq sim_L(u_p, u_q)$

Proof If $\forall u_p, u_q \in U, w_{u_p,u_q} = sim'_L(u_p, u_q) > 0$. There is at least one path from u_p to u_q, (u_p, u_q). The minimalhop distance of the path (u_p, u_q),

$$minimalhop_i(u_p, u_q) = w_{u_p,u_q} = sim'_L(u_p, u_q), \quad 1 \leq i \leq k,$$

$$sim_G(u_p, u_q) = maximinhop(u_p, u_q) \max_{k=1,\ldots,K} minimalhop_k(u_p, u_q) \geq w_{pq} = sim'_L(u_p, u_q).$$

If $\forall u_p, u_q \in U, w_{u_p,u_q} = 0$, then $sim_L(u_p, u_q) \leq 0$. From Proposition 1, $sim_G(u_p, u_q) \geq sim'_L(u_p, u_q)$. □

That is the global user similarity is non-negative and not less than the local user similarity. In addition, if the global user similarity between u_p and u_q is larger than their local similarity, it means there exists a path between them, along which any consecutive pair of nodes have larger local user similarity than $sim_L(u_p, u_q)$. In other words, two users become more similar because they can be connected through some locally more similar neighbors.

The Floyd-Warshall algorithm can be adopted to effectively compute all-pairs maximin distances (Aho and Hopcroft 1974; Cormen et al. 1992). The complexity of this algorithm is $O(N^3)$. In practice, an efficient algorithm (Kim and Choi 2007) based on message passing could be used to query the global similarity between a specific user u_* and the rest users, which exhibits a time complexity of $O(N^2)$.

Previous work (Fouss et al. 2007; Gori and Pucci 2007) has investigated global similarity measures for collaborative filtering. In (Fouss et al. 2007) and (Gori and Pucci 2007), collaborative filtering has been modeled as a bipartite graph, where nodes are users and items. These algorithms are random-walk based scoring algorithms, which can be used to rank items according to the active user's preferences rather than to predict his/her explicit ratings on items. However, our method aims to quantify the active user preferences; in a result it provides more information to recommendation systems than those just ranking items based on the active user's preferences.

4 The collaborative filtering framework

Taking both local and global users similarity into account, we propose the following collaborative filtering framework. To predict an active user's (u_a) rating on a particular item, under local similarity and global similarity we first find his k nearest neighbors both for the k local nearest neighbors ($nn_L^k(u_a)$) and the k global nearest neighbors ($nn_G^k(u_a)$). Then we employ both $nn_L^k(u_a)$ and $nn_G^k(u_a)$ to predict the user's rating

$$\hat{r}_{a,i} = (1-\alpha)\frac{\sum_{u_k \in nn_L^k(u_a)} sim'_L(u_k, u_a) r_{k,i}}{\sum_{u_k \in nn_L^k(u_a)} sim'_L(u_k, u_a)} + \alpha \frac{\sum_{u_k \in nn_G^k(u_a)} sim_G(u_k, u_a) r_{k,i}}{\sum_{u_k \in nn_G^k(u_a)} sim_G(u_k, u_a)}. \tag{1}$$

The parameter α determines the extent to which the prediction relies on local user similarity and global user similarity. With $\alpha = 0$, it indicates that the prediction depends completely on local user similarity and with $\alpha = 1$, it states that the prediction depends completely on global user similarity. α can be determined experimentally by using cross-validation.

5 Experiments

We conducted several experiments to examine the performance of the proposed collaborative filtering framework (LU&GU), and address the following questions in particular:

(1) How does our approach of computing the local user similarity compare with traditional methods? For this question, we employ PCC (Pearson Correlation Coefficient) (Resnick et al. 1994), PCCS (Pearson Correlation Coefficient with significance weighting) (Ma et al. 2007), SVS (surprisal-based vector similarity) and SVSS (surprisal-based vector similarity with significance weighting) as different methods to compute user similarity. Then we use these similarities in the traditional *user-based* collaborative filtering (Resnick et al. 1994) and compare the performance.

(2) How does our collaborative filtering framework compare with other algorithms? For this question, we compare our method (LS&GS) with the user-base algorithm (Resnick et al. 1994), the item-base algorithm (Sarwar et al. 2001), the similarity fusion algorithm (SF) (Wang et al. 2006) and the effective missing data prediction algorithm (EMDP) (Ma et al. 2007).

(3) How does the parameter α affect the accuracy of prediction? Parameter α balances how much the prediction takes into account local similarity and global similarity. We vary the value of α from 0 to 1 to observe the differences in performance.

5.1 Experimental setup

We experimented with a popular database, the MovieLens[1] dataset by the GroupLens Research group at University of Minnesota. The MovieLens data set contains 100,000 ratings (1–5 scales) from 943 users on 1682 movies (items), where each user has rated at least 20 movies.

To compare algorithms more thoroughly, we conducted the experiments under several configurations. We randomly exacted a subset of 500 users, altered the training size to be 300 (200, 100) users in the subset, and used the remaining 200 (300, 400) users

[1] http://www.grouplens.org/.

as the active users. The respective sets were named MovieLens300, MovieLens200 and MovieLens100. As for the ratings from the active users, we varied the number of ratings provided by the active users from 5, 10, and 20, naming them Given5, Given10 and Given20, respectively. This results in 9 configurations in total, which we call M300G20, M300G10, M300G5, M200G20, M200G10, M200G5, M100G20, M100G10 and M200G5. Different configuration represents different training data sparsity and test item of active user sparsity. These protocols are widely adopted (Wang et al. 2006; Ma et al. 2007; Xue et al. 2005). Furthermore, we also adopted the protocol of “All-but-one” (Breese et al. 1998), within which we extracted a single randomly selected rating for each user in the whole data set, and tried to predict its value given all the other ratings the user has voted on. The protocol is also widely adopted (Marlin 2004a, 2004b; DeCoste 2006).

In order to examine the performance of our approach and to compare it with experiments reported in the literature, e.g. (Resnick et al. 1994; Wang et al. 2006; Sarwar et al. 2001; Ma et al. 2007), we adopted the mean absolute error (MAE) (Sarwar et al. 2001). The MAE is computed by first summing the absolute errors of the N corresponding ratings-prediction pairs and then averaging the sum. Formally,

$$MAE = \frac{\sum_{i=1}^{N} |r_i - \hat{r}_i|}{N}.$$

A smaller value of MAE indicates a better accuracy.

5.2 Surprisal-based vector similarity

In order to examine the performance of SVSS and SVS, we compared our methods of computing the user similarity with other traditional methods, PCC and PCCS. We used these methods in the traditional *user-based* collaborative filtering and compared their performance. The parameter γ (used in SVSS and PCCS) of the significance weighting was set to 20.

We compared SVSS and SVS with other methods in all experimental configurations. The number of nearest neighbors in *user-based* collaborative filtering was set as 35 in all configurations.

The results are presented in Table 1 and Table 2. We can see that:

(1) SVSS and SVS outperform the other methods in all configurations.
(2) The performance of SVSS and SVS improves with the number of items rated by the users.
(3) Significance Weighting improves SVS much more than PCC except in the All-but-one protocol. The reason for that is SVS can get more accurate contributions of each rating to the value of similarity than PCC does. Using significance weighting amplifies the influence.

Next, in order to examine the sensitivity of the neighborhood size, we performed an experiment where we varied the number of nearest neighbors that were used and computed the MAE for each variation. In this article, we report only the results for the configurations M100G5 and M300G20, however, the other configurations yield similar results. The results are shown in Fig. 1.

We can observe that the size of neighborhood does affect the performance. Both SVS and SVSS improve the accuracy of prediction as the neighborhood size increases from 5 to 15. For greater values, the curve flattens. Again, SVS and SVSS outperform the other methods.

Table 1 MAE comparison of different methods of computing the user similarity on the MovieLens dataset (the smaller the value, the better the performance)

Training users	Methods	Given5	Given10	Given20
100	SVSS	**0.8173**	**0.7843**	**0.7743**
	SVS	0.8232	0.7914	0.7813
	PCCS	0.8335	0.8004	0.7918
	PCC	0.8377	0.8044	0.7934
200	SVSS	**0.814**	**0.7908**	**0.7792**
	SVS	0.8193	0.7995	0.7931
	PCCS	0.8156	0.7995	0.7954
	PCC	0.8185	0.8067	0.7960
300	SVSS	**0.784**	**0.7786**	**0.7407**
	SVS	0.7883	0.7874	0.7564
	PCCS	0.8040	0.7865	0.7689
	PCC	0.8055	0.7910	0.7805

Table 2 MAE comparison of different methods of computing the user similarity on the MovieLens dataset (the smaller the value, the better the performance)

Methods	SVSS	SVS	PCCS	PCC
All-but-one	**0.72**	0.7232	0.7466	0.7625

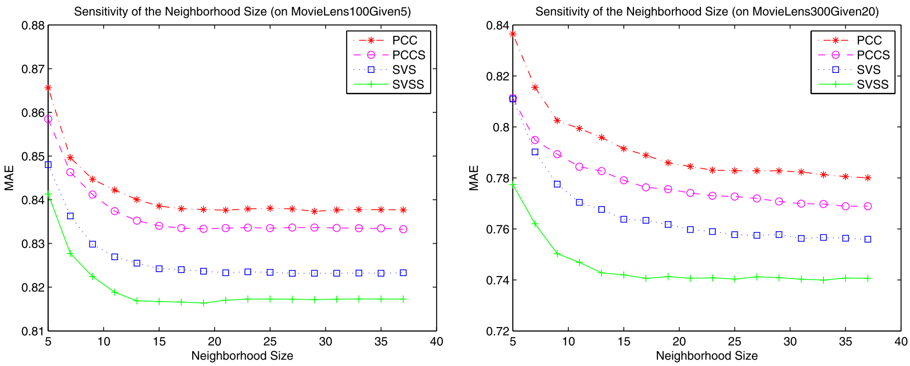

Fig. 1 (**a**) Sensitivity of the neighborhood size (on M100G5). (**b**) Sensitivity of the neighborhood size (on M300G20)

In addition, it can be seen that SVS and SVSS results in much better performance than other methods when there are more ratings (Given20) from active users in the training data. The reason is that for computing the similarity, SVS and SVSS have access to more accurate contributions from each item.

5.3 Comparison of our framework of collaborative filtering and other methods

We compared the following algorithms: the *user-based* using PCC (UPCC) (Resnick et al. 1994), the *item-based* methods (IPCC) (Sarwar et al. 2001), the similarity fusion algorithm

Table 3 MAE comparison with state-of-the-arts algorithms on MovieLens (A smaller value means a better performance)

Training users	Methods	Given5	Given10	Given20
100	LU&GU	0.791	0.7681	**0.7565**
	EMDP	**0.7896**	**0.7668**	0.7806
	SF	0.8446	0.7807	0.7717
	UPCC	0.8377	0.8044	0.7943
	IPCC	0.9639	0.8922	0.8577
200	LU&GU	**0.7937**	**0.7733**	**0.7719**
	EMDP	0.7997	0.7953	0.7908
	SF	0.8507	0.8012	0.7862
	UPCC	0.8185	0.8067	0.796
	IPCC	0.955	0.9135	0.871
300	LU&GU	**0.7718**	**0.7704**	**0.7444**
	EMDP	0.7925	0.7951	0.7552
	SF	0.8062	0.7971	0.7527
	UPCC	0.8055	0.7910	0.7805
	IPCC	0.9862	0.9266	0.8573

Table 4 MAE comparison with state-of-the-arts algorithms on MovieLens (A smaller value means a better performance)

Methods	LU&GU	EMDP	SF	UPCC	IPCC
All-but-one	**0.719**	0.8017	0.7413	0.7625	0.7919

(SF) (Wang et al. 2006) and the effective missing data prediction (EMDP) algorithm (Ma et al. 2007). The parameters of SF were set to $\lambda = \delta = 0.4$, $k = 35$. The parameters of EMDP were set to $\lambda = 0.6$, $\gamma = 30$, $\delta = 25$, $\eta = \theta = 0.6$. The parameters of our method were set to $\gamma = 30$, $k = 35$, $\alpha = 0.5$. Table 3 and Table 4 summarized our results. Our method outperforms UPCC, IPCC and SF in all configurations and outperforms EMDP in the most configurations. In the conditions Movie100Given10 and Movie100Given5, our results are very close to EMDP's. We want to point out that EMDP is a combination of a *user-based* predictor and an *item-based* predictor. Our approach in nature is an improvement of *user-based* algorithms. Hence our method can be easily employed by EMDP to replace the traditional *user-based* approaches and achieve a better performance.

5.4 Impact of parameter

As discussed above, we employed the parameter α in Eq. 1 to balance the prediction from local user similarity and the prediction from global user similarity. Next, in order to determine the sensitivity of the parameter α in Eq. 1, we carried out several experiments on all configurations in which we varied the value of α from 0 to 1, iteratively incrementing it by 0.05. The results are shown in Figs. 2 and 3.

With $\alpha = 0$, the prediction depends completely on local user similarity and with $\alpha = 1$, the prediction depends completely on global user similarity. Figures 2(a) and 3(a) shows that when there exist few ratings of the active users or few training users, the global user similarity will help to improve the prediction accuracy in a great deal. But when there exist plenty of ratings from the active users and more training users. Then the global similarity cannot obviously improve the accuracy. This can also be observed in Figs. 2(b) and 3(b).

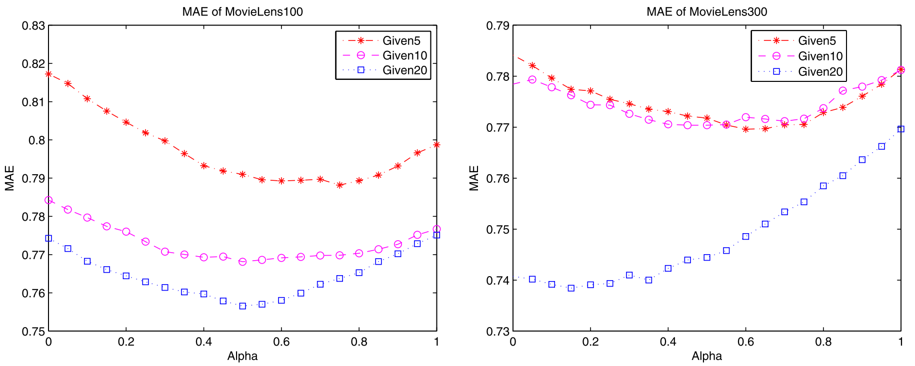

Fig. 2 (**a**) Impact of Alpha on MAE (on MovieLens100). (**b**) Impact of Alpha on MAE (on MovieLens300)

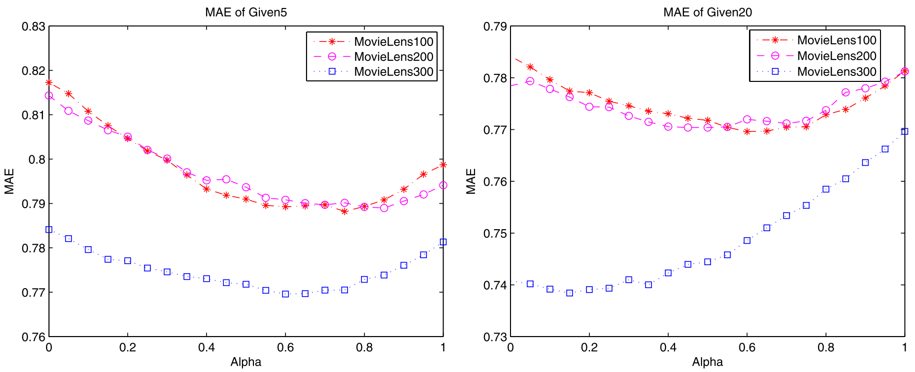

Fig. 3 (**a**) Impact of Alpha on MAE (on Given5). (**b**) Impact of Alpha on MAE (on Given20)

6 Conclusions

In this paper, we first proposed to describe the relationship of users using local user similarity and global user similarity. Then we proposed new methods to compute local user similarity and global user similarity, and a collaborative filtering framework based on both of these user similarity measures. SVS (local user similarity) considers the quantities of information (surprisal) of each rating to determine the contribution of any two ratings of two users to calculate their similarity. The intuition behind this method is that less common ratings for a specific items trend to provide more discriminative information than the most common ones. Under global user similarity, two users become more similar if they can be connected through their locally similar neighbors. The proposed collaborative filtering framework employs both local and global user similarity to make rather accurate predictions. Experimental results show: (1) using SVS (local user similarity) can find high quality neighbors; (2) our proposed framework (LU&GU) can improve the accuracy of predication; (3) under the sparse data set condition, the global user similarity can improve the performance of the algorithm which uses only local user similarity.

In the future, we plan to investigate how to combine local and global user similarity in a more natural way. Furthermore, we have started to investigate whether it is possible to incorporate information about items in the proposed framework.

Acknowledgements The authors would like to thank Darong Lai and Xiangyang Liu for helpful discussions. This work was partially supported by National Natural Science Foundation of China under award No.60672066 and National High-tech Research and Development Program of China under Grant No. 2007AA01Z157.

References

Aho, A. V., & Hopcroft, J. E. (1974). *The design and analysis of computer algorithms*. Boston: Addison-Wesley.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the fourteenth annual conference on uncertainty in artificial intelligence* (pp. 3–52).

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1992). *Introduction to algorithms*. Cambridge: MIT Press.

DeCoste, D. (2006). Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proceedings of the 23rd international conference on machine learning* (pp. 249–256).

Fouss, F., Pirotte, A., Renders, J. M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 355–369.

Gori, M., & Pucci, A. (2007). ItemRank: a random-walk based scoring algorithm for recommender engines. In *IJCAI* (pp. 2766–2771).

Herlocker, J. L., Konstan, J. A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 230–237).

Hofmann, T., & Puzicha, J. (1999). Latent class models for collaborative filtering. In *Proceedings of the sixteenth international joint conference on artificial intelligence* (pp. 688–693).

Jin, R., Chai, J. Y., & Si, L. (2004). An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th annual international conference on research and development in information retrieval* (pp. 337–344).

Kim, K. H., & Choi, S. (2007). Neighbor search with global geometry: a minimax message passing algorithm. In *Proceedings of the 24th international conference on machine learning* (pp. 401–408).

Kotz, S., Kozubowski, T. J., & Podgrski, K. (2001). *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Basel: Birkhauser.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Transactions on Internet Computing*, *7*(1), 76–80.

Ma, H., King, I., & Lyu, M. R. (2007). Effective missing data prediction for collaborative filtering. In *SIGIR '07: proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 39–46). New York: ACM.

Marlin, B. (2004a). Modeling user rating profiles for collaborative filtering. *Advances in Neural Information Processing Systems*, *16*, 627–634.

Marlin, B. (2004b). *Collaborative filtering: a machine learning perspective*. Master thesis, University of Toronto.

Norton, R. M. (1984). The double exponential distribution: Using calculus to find a maximum likelihood estimator. *The American Statistician*, *38*(2), 135–136.

Pennock, D. M., Horvitz, E., Lawrence, S., & Giles, C. L. (2000). Collaborative filtering by personality diagnosis: a hybrid memory-and model-based approach. In *Proceedings of the 16th conference on uncertainty in artificial intelligence* (pp. 473–480).

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on computer supported cooperative work* (pp. 175–186).

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285–295).

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 210–217).

Soboroff, I., & Nicholas, C. (2000). Collaborative filtering and the generalized vector space model (poster session). In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 351–353).

Wang, J., de Vries, A. P., & Reinders, M. J. T. (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* (pp. 501–508).

Xue, G. R., Lin, C., Yang, Q., Xi, W. S., Zeng, H. J., Yu, Y., et al. (2005). Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 114–121).