Knowledge-Based Systems xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Exploiting matrix factorization to asymmetric user similarities in recommendation systems

Parivash Pirasteh [a], Dosam Hwang [a,1], Jason J. Jung [b,*,1]

[a] Department of Computer Engineering, Yeungnam University, Republic of Korea
[b] Department of Computer Engineering, Chung-Ang University, Republic of Korea

ARTICLE INFO

Article history:
Received 15 September 2014
Received in revised form 3 February 2015
Accepted 6 March 2015
Available online xxxx

Keywords:
Collaborative filtering
Matrix factorization
Recommender systems
User similarity
Asymmetry

ABSTRACT

Although collaborative filtering is widely applied in recommendation systems, it still suffers from several major limitations, including data sparsity and scalability. Sparse data affects the quality of the user similarity measurement and consequently the quality of the recommender system. In this paper, we propose a novel user similarity measure aimed at providing a valid similarity measurement between users with very few ratings. The contributions of this paper are twofold: First, we suggest an asymmetric user similarity method to distinguish between the impact that the user has on his neighbor and the impact that the user receives from his neighbor. Second, we apply matrix factorization to the user similarity matrix in order to discover the similarities between users who have rated different items. Experimental results show that our method performs better than commonly used approaches, especially under cold-start condition.

1. Introduction

In the age of data overload, the enormous amount of information makes users to spend more time and energy to select an item. The item can be a book for an Amazon user or a place to visit for a tourist or a course to pass for a learner. In order to overcome information overload problem, recommender systems aid users to find their desired content in a reasonable time, by analyzing their behavior data related to user activity [20].

Content-based filtering is based on the hypothesis that users are able to formulate queries that express their interests or information needs in terms of intrinsic features of the desired items. The user profile is a structured representation of user interests, extracted to recommend new items. The recommendation process basically consists of matching the characteristics of the user profile against the characteristics of a content object [14,22].

Collaborative filtering (CF) is an alternative to content-based techniques. Instead of recommending new items that are similar to items the user has liked in the past, this method recommends items that similar users have liked [2,3]. CF techniques are more often implemented than content filtering and often result in better predictive performance. The main reason is that they are independent of data used by content filtering, which are invasive and time consuming to collect. Generally, CF outperforms content-based techniques except in special cases, such as when user ratings of a certain item are highly varied (i.e. controversial items) or for cold-start situations, where the users did not provide enough ratings to compute similarity with other users [30].

A key factor in the quality of the recommendations obtained in a CF-based recommender system lies in its capacity to determine which users have the most in common with a given user. Traditional methods of similarity suffer from three drawbacks. First, usually all items are treated equally, regardless of the various amounts of information that can be extracted from different items. This is addressed by [16], where weighting schema are defined in order to capture the importance of an item and giving distinct items a higher coefficient in assigning a correlation.

The second problem occurs under cold-start condition where recommender systems need to predict preferences for a user with a small number of ratings. Pure CF methods which rely on calculating similarities between users based on their co-rated items, fail to provide similarity between users who do not have any co-rated items. To illustrate this limitation, consider the example of Fig. 1. While John and Alex share the same neighbors, the similarity coefficient between them cannot be computed based on traditional methods because they do not have any common item. However, content information can help bridge the gap between existing items and new items by inferring similarities among them [32]. Several hybrid methods (combinations of content-based and CF techniques) have been proposed to improve the performance of

* Corresponding author. Tel.: +82 2 820 5316; fax: +82 2 820 5301.
E-mail address: j2jung@gmail.com (J.J. Jung).
[1] These authors contributed equally to this work as the first author.

http://dx.doi.org/10.1016/j.knosys.2015.03.006
0950-7051/© 2015 Elsevier B.V. All rights reserved.

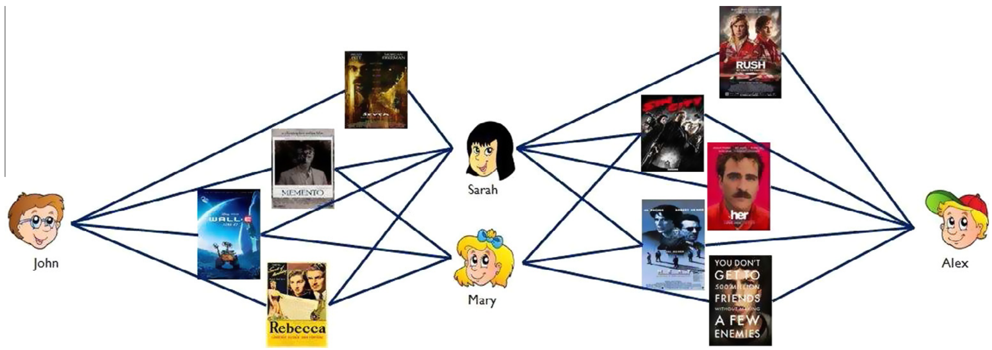


Fig. 1. Although John and Alex do not have a direct overlap between their rated items, the similarity ratio between them and their joint neighbors can express their agreement on and disagreement over on different movies.

recommender systems when it comes to cold-start prediction. Although content-based methods are not very sensitive to data sparsity, they suffer from other limitations. For example, the content information is hard to automatically extracted out, and is not always available for privacy reasons.

The proposed method in the paper allows generating high quality recommendations even on sparse dataset. It also helps to learn dependencies between all the pairs of users by projecting the existing similarity values in a latent space model. In this paper, first, we propose an asymmetric user similarity measurement based on mean square difference and cosine similarity. In this way, we incorporate into the proposed model the ability to distinguish between two users with a different proportion of common ratings. Second, we apply matrix factorization to characterize the user's interests by a vector of factors derived from the proposed similarity measure in order to predict similarity among users with few immediate neighbors.

The rest of this paper is organized as follows. In Section 2, we present some of the most relevant works on the topic and describe advantages and disadvantages. Section 3 presents our proposed method for user similarity measurement. The user similarity matrix factorization framework is presented in Section 4. The results of experimental analysis are presented in Section 5, followed by the conclusion and future directions in Section 6.

2. Related works

The most commonly used measurement techniques for similarities between users are the Pearson correlation coefficient (PCC) [29] and cosine similarity (COS) algorithm [5]. PCC defines user similarity as the linear correlation between them. It is well recognized that PCC and COS only consider the direction of rating vectors and ignore the length [26].

COS assumes that the rating of each user is a point in a vector space and then evaluates the cosine angle between the two points. It considers the common rating vectors $X = \{x_1, x_2, x_3, \ldots, x_n\}$ and $Y = \{y_1, y_2, y_3, \ldots, y_n\}$ of users X and Y, represented by a dot product divided by magnitude. COS has frequently been used for performance comparisons in CF [19]. The constrained Pearson correlation coefficient is a slightly modified version of the Pearson correlation that increases the correlation only when both users have rated an item positively or negatively [33].

Although the two popular similarity measures, PCC and COS have proven to be successful in many studies, they have some drawbacks. The main limitation of these approaches is their inability to consider the size of the set of common items between users. To overcome this limitation, a combination of Jaccard similarity with Pearson correlation coefficient has been proposed [34].

Jaccard similarity does not suffer from this limitation because it measures the overlap that two vectors share with their attributes. On the other hand, such a measure does not take into account the difference in ratings between the vectors. In this case, if two users watch the same movies but have completely opposite opinions of them, the users are considered to be similar regardless of their differing opinions [6].

Like PCC, the Spearman rank is correlation coefficient based and computes a measure of correlation between ranks instead of actual preference scores [36]. Shardanand et al. proposed a measure based on mean square difference (MSD), which evaluates the similarity between two users as the inverse of the average squared difference between the ratings given by those users on the same item [33].

Other experts proposed new similarity measures to substitute for the traditional one. Konstan et al. suggested a concordance-based measure that helps users with privacy concerns and those who do not want to reveal their ratings history [17]. On the data sparsity problem, one noteworthy study presented a similarity method using proximity-impact-popularity (PIP). Ahn discussed the problems of widely used similarity methods resulting in decreased prediction accuracy and proposed PIP similarity [1]. Using genre information to circumvent the cold-start problem for new items has been used in [28]. Hence, when a new item enters the system, genre correlation aids in finding similar items.

Luo et al. [23] divide user similarity into two parts: local user similarity and global user similarity. Local similarity is determined based on surprisal-based vector similarity (SVS). Global similarity measures the similarity between two users by further considering the extent to which their neighbors are locally similar (using the local similarity). Therefore, two users become more similar if they can be connected through a series of locally similar neighbors. From this point of view, the rest of the mentioned methods are categorized as global similarity measures.

Moreover, Jamali et al. introduced a similarity measure using the Markov-chain model of a random walk. This approach can weaken the similarity of small common items among users [15]. In graph-based approaches, the data are represented in the form of a graph, where nodes are users, items or both, and edges encode the interactions or similarities among the users and items. The transitive associations captured by graph-based methods can be used to recommend items. Fouss et al. suggested applying Euclidean commute time distance, which is one of the random-walk-based methods to compute similarities between nodes. The SimRank algorithm is another graph-based model that is utilized to compute the similarity [12]. For example [11] applied the SimRank-based algorithm to construct a general recommendation system, while [7] proposed combining SimRank with clustering to match users in online dating networks.

Please cite this article in press as: P. Pirasteh et al., Exploiting matrix factorization to asymmetric user similarities in recommendation systems, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.006

In path-based similarity, the distance between two nodes of the graph is evaluated as a function of the number of paths connecting the two nodes, as well as the length of these paths. The number of paths between a user and an item in a bipartite graph can also be used to evaluate their compatibility [13].

Finally, dimensionality reduction technique (e.g., matrix factorization) is commonly used to reduce the dimensionality of the recommender system database. Matrix factorization is especially adequate for processing large recommender system databases and providing scalable approaches [4,24].

3. The asymmetric user similarity model

Although, many similarity measurements have been developed to make up for some potential weaknesses under traditional similarity, these methods still suffer from some shortcomings. Most of the similarity metrics assign equal value for the similarity relation between two users, This means, these methods are based on the assumption that $\mathrm{sim}(u,v) = \mathrm{sim}(v,u)$. Here we use $\mathrm{sim}(u,v)$ to show the similarity between user u and user v.

Table 1 gives a hypothetical example of a user profile including user rating information for different items. From Table 1 we can see that $user_1$ ratings are quite similar to $user_2$, but not vice versa. Traditional methods cannot differentiate between these two users with different rating profiles. Therefore, the impact that $user_1$ receives in predicting a new item's rating from $user_2$ is equal to the impact that $user_2$ receives from $user_1$ for recommending a new movie.

To avoid this contradiction, we use an asymmetric similarity measure, which is defined as the proportion of co-rated items between users, normalized by the number of items rated by active user.

$$\mathrm{sim}(u,v) = \frac{|I_u \cap I_v|}{|I_u|} \tag{1}$$

where I_u and I_v represent the set of items rated by users u and v, respectively. Eq. (1) only considers the ratio of common ratings that users possess among all their rated items and ignores the proportion of common ratings in total number of ratings between users. Hence, an additional parameter is needed to combine with Eq. (1). This second coefficient is referred as the Sorensen index.

$$\mathrm{sim}(u,v) = \frac{|I_u \cap I_v|}{|I_u|} \cdot \frac{2 * |I_u \cap I_v|}{|I_u| + |I_v|} \tag{2}$$

On the other hand, discarding the absolute value of rating leads to very low accurate similarity measurements. To consider the absolute rating in the formula, we propose to combine Eq. (2) with other similarity measures, in order to benefit from their respective advantage points. A combination of Eq. (2) with COS similarity is the first proposed method which is expressed as:

$$\mathrm{ACOS}(u,v) = \frac{\overrightarrow{r_u} \cdot \overrightarrow{r_v}}{\|\overrightarrow{r}_u\| \cdot \|\overrightarrow{r}_v\|} \cdot \frac{|I_u \cap I_v|}{|I_u|} \cdot \frac{2 * |I_u \cap I_v|}{|I_u| + |I_v|} \tag{3}$$

Table 1
An example of a user-item matrix.

	Item1	Item2	Item3	Item4	Item5	Item6
user1	4		2			
user2	4	1	2	1	1	1
user3		2		3		
user4		1	2			
user5	4				4	4
user6		1	2	2	1	1

The second suggested similarity measure combines Eq. (2) with MSD [33]. MSD computes the similarity between users based on the mean difference of common rated items, by using the following formula:

$$\mathrm{MSD}(u,v) = \frac{\sum_{p \in |I_u \cap I_v|} (r_{u,p} - r_{v,p})^2}{|I_u \cap I_v|} \tag{4}$$

The similarity measure based on MSD can be formulated as follows:

$$\mathrm{sim}(u,v) = \frac{L - \mathrm{MSD}(u,v)}{L} \tag{5}$$

Users where the difference is greater than a certain threshold, L, are discarded. To keep maximum similarities information between users we set the threshold L to 16, which is the maximum possible MSD value between two users. Our second user similarity method can be written as follows:

$$\mathrm{AMSD}(u,v) = \frac{L - \mathrm{MSD}(u,v)}{L} \cdot \frac{|I_u \cap I_v|}{|I_u|} \cdot \frac{2 * |I_u \cap I_v|}{|I_u| + |I_v|} \tag{6}$$

Eq. (2) is used as a weighting factor. Because it made the similarity measures asymmetric, the new similarity measures are called asymmetric COS (ACOS) and asymmetric MSD (AMSD). Table 2(a) and (c) give user similarity matrices according to AMSD and ACOS, respectively. We can see that $\mathrm{sim}(user_1, user_2)$ is higher than $\mathrm{sim}(user_2, user_1)$. This is because the $user_1$ rated items are a subset of the $user_2$ rated items but not vice versa.

The proposed method not only has the advantage of distinguishing between two users similarities but also provides more accurate similarities between multiple users. From Table 1 we can see that the $\mathrm{sim}(user_2, user_6)$ should be higher than $\mathrm{sim}(user_2, user_1)$, whereas the traditional MSD and COS do not recognize this assumption. This is because MSD and COS ignore the proportion of common ratings.

4. User similarity matrix factorization

The basic idea behind matrix factorization approaches is to fit the original user-item rating matrix with a low-rank approximation, one containing the so-called "user factors", and the other containing the so-called "item-factors". Thus, each user u is represented with an f-dimensional user factors vector $p_u \in R^f$. Similarly, each item i is represented with an item factors vector $q_u \in R^f$ [9,25,37]. In our approach, we factorize a user-user matrix containing ACOS and AMSD similarities between pairs of users to obtain the latent features. Let us consider the user similarity matrix in Table 2(a) and (c). There are 6 users in total. The similarity between users who do not share any items is indicated with an unknown value (*). The main target is prediction of the missing values of the user similarity matrix. Let $S = \{S_{i,j}\}$ denote an m*m matrix of user similarity, where $S_{ij} \in (0,1]$ denotes the similarity between user i and j based on ACOS and AMSD, and m is the total number of users.

Matrix factorization represents the user-user similarity matrix S with two low-rank matrices, $U \in R^{k\times m}$ and $Z \in R^{k\times m}$. U_i and Z_j are the column vectors and indicate K-dimensional latent feature vectors of user i and j, respectively. We can define the optimization function as follows:

$$L(U,Z) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} I_{ij}^{R}(S_{i,j} - U_i^T Z_j)^2 + \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|Z\|_F^2 \tag{7}$$

where $\lambda_1, \lambda_2 > 0$, $\|\cdot\|_F$ denotes the Frobenius norm and I_{ij}^R is the indicator function that is equal to 1 if user i and j have a similarity ratio based on ACOS or AMSD and is otherwise equal to 0. The

Table 2
A user similarity matrix.

	u1	u2	u3	u4	u5	u6
*(a) User similarity matrix based on AMSD. The missing similarity values are represented by **						
u1	–	0.33	*	0.16	0.12	0.08
u2	0.11	–	0.09	0.11	0.18	0.68
u3	*	0.28	–	0.15	*	0.37
u4	0.16	0.33	0.15	–	*	0.4
u5	0.08	0.37	*	*	–	0.13
u6	0.03	0.82	0.15	0.16	0.08	–
(b) Predicted user similarity matrix using matrix factorization						
u1	–	0.33	0.05	0.16	0.12	0.08
u2	0.11	–	0.09	0.11	0.18	0.68
u3	0.1	0.28	–	0.15	0.09	0.37
u4	0.16	0.33	0.15	–	0.12	0.4
u5	0.08	0.37	0.05	0.04	–	0.13
u6	0.03	0.82	0.15	0.16	0.08	–
*(c) User similarity matrix based on ACOS. The missing similarity values are represented by **						
u1	–	0.3	*	0.06	0.06	0.02
u2	0.1	–	0.03	0.05	0.18	0.38
u3	*	0.09	–	0.04	*	0.26
u4	0.06	0.15	0.04	–	*	0.26
u5	0.04	0.36	*	*	–	0.07
u6	0.009	0.46	0.1	0.1	0.04	–
(d) Predicted user similarity matrix using matrix factorization						
u1	–	0.3	0.03	0.06	0.06	0.02
u2	0.1	–	0.03	0.05	0.18	0.38
u3	0.06	0.09	–	0.04	0.05	0.26
u4	0.06	0.15	0.04	–	0.01	0.26
u5	0.04	0.36	0.1	0.1	–	0.07
u6	0.009	0.46	0.1	0.1	0.04	–

optimization aims to approximate the observed value $S_{i,j}$ by $U_i^T Z_j$, a product of two low-rank vectors, with regularization U and Z. Therefore, the unknown values can be predicted by minimizing the objective function, Eq. (7), using gradient descent, with alternatively fixed U and V.

Although, gradient descent has proven to be an effective way of factorizing the matrix, it usually suffers from slow convergence which makes the algorithm time consuming as the number of iterations is often quite large. By applying MF on user-item rating matrix, the sparsity of matrix helps to reduce the number of iterations during run time. However, in most cases, the similarity matrix S is dense and the gradient descent requires multiple iterations to converge. Several methods have been proposed to break down the computational time of gradient descent. For example, parallel computing can be employed to speed up the gradient descent by computing it over multiple instances in parallel. Since the gradient is a sum training instances, it is possible to split the computation into different blocks [8]. The communication overhead can be reduced by updating the parameter vectors asynchronously [21]. The key advantages of an asynchronous framework is that it allows more frequent parameter updates, a more flexible design and a more robust system to individual node failures. Alternative techniques have also been suggested in [35,10,27] which can be applied for improving the speed of convergence of gradient descent learning algorithms.

After finding the similarities between all users, a weighted aggregate of a neighbor's ratings will be used to estimate a prediction for the active user. This estimated value would be the rate that the user would give to an item, if he were to evaluate it. The predicted rating of item i for the active user a is the weighted sum of the ratings given to item i by the k neighbors of a, according to following formula:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{k} \mathrm{sim}(u,a) \cdot (r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{k} |\mathrm{sim}(u,a)|} \tag{8}$$

where $\bar{r}_a$ and $\bar{r}_u$ are the average ratings for user a and user u, and $\mathrm{sim}(u,a)$ is the similarity between user u and user a. The summations are over all the users who have rated the item i.

5. Experimental results

In order to prove the effectiveness of the proposed methods, several experiments were performed focusing on testing recommendation performance. This section provides the experimental results from the proposed methods.

5.1. Experimental setup

5.1.1. Datasets

We conduct the experiments using the MovieLens[2] and Netflix[3] datasets. 2000 users of each dataset have been selected randomly. The total number of ratings used for experiments is 950390 in MovieLens dataset and 658809 in Netflix dataset. The ratings scale ranges between 1- and 5, where 1 = bad and 5 = excellent. We randomly divided the ratings into two disjoint sets: training and test. The training set is utilized to calculate predictions using each algorithm. Data from the test set are used to assess the predictive capability of the model.

5.1.2. Metrics

To evaluate the prediction quality of the proposed methods, we employ two accuracy metrics: mean absolute error (MAE) and root mean squared error (RMSE).

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum\nolimits_{i,j}(R_{i,j} - \overline{R}_{i,j})^2} \tag{9}$$

$$\mathrm{MAE} = \frac{1}{T}\sum_{i,j}|R_{i,j} - \overline{R}_{i,j}| \tag{10}$$

Here R_{ij} denotes the rating user i gave to item j, $\overline{R}_{i,j}$ denotes the rating user i gave to item j as predicted by ACOS or AMSD methods, and T denotes the number of tested ratings. A smaller value of MAE or RMSE signies better prediction quality.

5.2. Impact of dimensionality K on the results

The impact of a different number of latent features on prediction error is observed in this experiment. Fig. 2 illustrates the accuracy of the suggested methods on the MovieLens dataset and Netflix dataset. We can notice that accuracy improves on MovieLens dataset when the number of latent features increases to 5 and decreases when the number of latent features increases to 10. The optimum value is reached when dimensionality = 5. This can be interpreted to mean that increasing dimensionality to 5 provides a more distinct set of parameters and results in better accuracy but increasing dimensionality to 10 causes overfitting. The same conclusion can be derived on Netflix, with a different optimum dimensionality. Both methods show their best performance with dimensionality = 7. From the results we observe that no matter which dataset is used, by increasing the dimensionality to a large number the RMSE value increases, and consequently the accuracy of method decreases. Accordingly, in all the rest of the experiments the dimensionality is 7 for the Netflix dataset and 5 for the MovieLens dataset. We also repeat the experiment to find the optimal dimensionality value for other MF based methods compared in this study including SVD and MF. The results suggest

[2] http://www.grouplens.com.
[3] http://www.netflixprize.com.

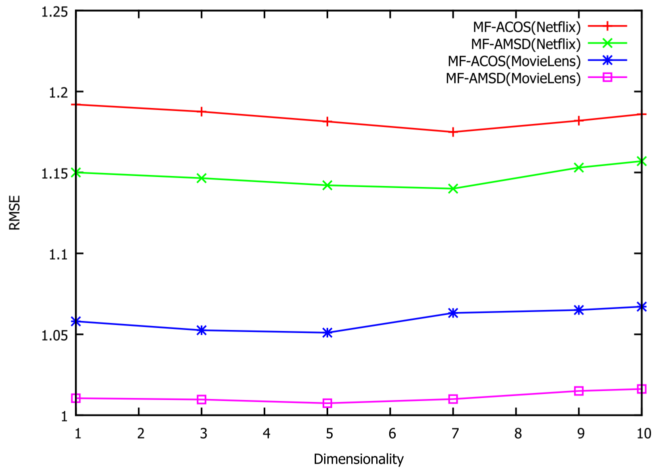


Fig. 2. Dimensionality analysis.

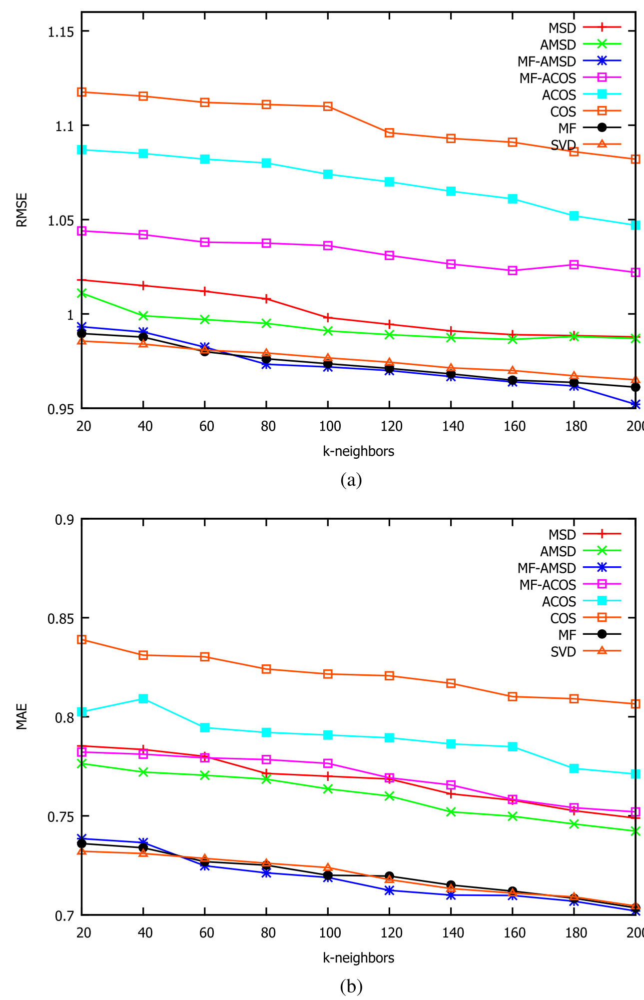


Fig. 3. The performance of different similarity measures with different k-neighbors on MovieLens dataset.

that the best dimensionality values are 6 and 8, for SVD and MF respectively.

5.3. The effect of the number of nearest neighbor

In order to indicate the effect of the number of nearest neighbors on recommendation accuracies, this experiment was performed using both datasets (Netflix and MovieLens). We compare the performance of the proposed algorithms with COS and MSD methods and methods based on matrix factorization (MF, SVD). MF algorithm is basic state-of-the-art matrix factorization technique which is introduced in [18] and SVD is a classical dimensionality reduction-based algorithm which decomposes the user-item matrix into three matrices [31]. Figs. 3 and 4 show the error rates of different similarity measures with different numbers of nearest neighbors on the MovieLens and Netflix datasets, respectively. As shown in Fig. 3(a), increasing the number of neighbors decreases the RMSE of all similarity measures. From the figure we can see that matrix factorization based on AMSD (MF-AMSD) can obtain the best performance when the number of neighbors k is more than 60. Comparing RMSE of AMSD and MF-AMSD, we observe an improvement (e.g. using 200 neighbors, RMSE decreases to 0.9521 from 0.987).

Fig. 3(b), illustrates the generated results for MAE. Same as RMSE, MAE confirms the superiority of MF-AMSD over other similarity measures.

Fig. 4(a) and (b) show the change in performance from different similarity measures with different K-neighbors on the Netflix dataset. As Fig. 4(a) shows, MF-AMSD consistently outperforms other approaches, so applying the MF-AMSD method to the Netflix dataset shows a significant improvement in results compared to when it is applied to the MovieLens dataset. An explanation could be that the sample data of the Netflix database are more sparse, so effectiveness of matrix factorization in predicting missing values is more tangible.

Fig. 4(b) shows that the same conclusions can be drawn by using MAE as a criterion for the test set. MF-AMSD improves the MAE of MSD method by 7.7% for $k = 200$. While MF-AMSD is the most reliable technique, MF-ACOS outperforms the remaining methods. In conclusion, our improved similarity measure can obtain better performance than other measures. Moreover, our method not only calculates the similarity measures for users with common rated items, but also provides similarity value for a pair of users without shared items. Although, the superiority of our method is not remarkable, it obtained the best RMSEs, MAEs with different number of nearest neighbors which is result of stability. In addition, by increasing number of neighbors on Netflix dataset the difference between performance of suggested method and other methods becomes more significant.

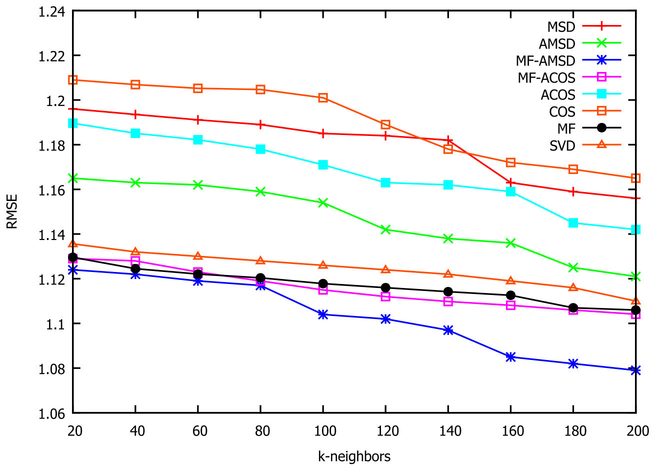
5.4. Comparison of performance on different groups of users

In order to examine how much user similarity matrix factorization contributes to the final recommendation, we assign two different values to unknown similarity values and compare the output result with the matrix factorization's predicted value. First, we assign a random value to any pair of users without a similarity value; second, we set all the unknown similarities to 1. We categorize users into three different subsets:

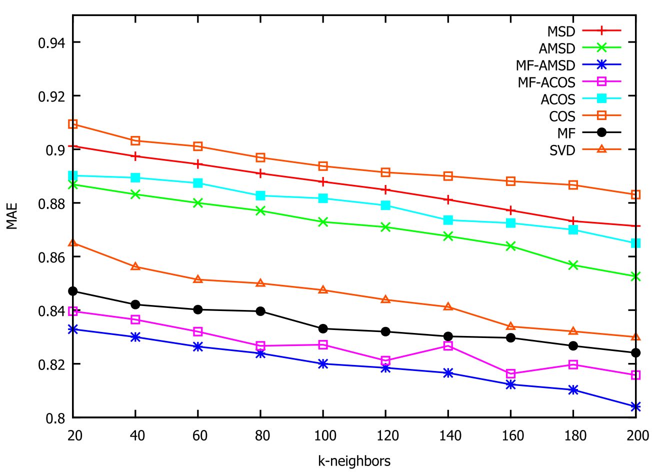
- $subset_1$: users who have a similarity value with fewer than 25% of other users.
- $subset_2$: users who have a similarity value with fewer than 50% and more than 25% of other users.
- $subset_3$: users who have a similarity value with more than 75% of other users.

The result of experiment 3 is shown in Table 3. We can see that AMSD that includes unknown similarity values to 1 or to random values has lower accuracy than MF-AMSD. This result demonstrates the positive influence of matrix factorization in predicting

Please cite this article in press as: P. Pirasteh et al., Exploiting matrix factorization to asymmetric user similarities in recommendation systems, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.006



(a)



(b)

Fig. 4. The performance of different similarity measures with different k-neighbors on Netflix dataset.

Table 3
Impact of matrix factorization on predicting similarity values for different subsets of users. The error rate is based on RMSE.

Dataset	User subset	MF-AMSD	AMSD (sim = Random)	AMSD (sim = 1)
MovieLens	$subset_1$	0.9716	1.034	1.039
	$subset_2$	0.9527	1.026	1.012
	$subset_3$	0.9431	0.994	0.99
Netflix	$subset_1$	1.112	1.181	1.178
	$subset_2$	1.086	1.157	1.145
	$subset_3$	1.062	1.142	1.13

similarity values. It can also easily be seen that a richer user similarity matrix leads to a better result.

6. Conclusion and future work

In this paper, we presented a similarity measurement based on an asymmetric structure that is calculated between each pair of users. Based on the assumption that more information about user similarity can lead to improvement in recommendations, we proposed user similarity matrix factorization to estimate the missing similarity values in the user-user similarity matrix. Our method provides a framework for computing similarity between pair of users from the similarities between their neighbors. This method assumes that similarity values are a linear combination of several latent features. The latent factors could be the extent of interest that a user has in different features of an item.

Through experiments performed on MovieLens and Netflix datasets, we conclude that MF-AMSD performs better in comparison to the traditional similarity measurements. The experimental results confirm that applying matrix factorization to user similarity computation provides more accurate results, and performance is more pronounced when the data are sparse.

With the increased use of social networking, most of the research topics will be directed to social-based research. We will seek to develop our framework in social networks. One future direction is to consider the relationship between users by employing a trust network. The connection between users in a trust network can be considered as a candidate factor affecting the similarity between them.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2014R1A2A2A05007154).

References

[1] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Inf. Sci. 178 (1) (2008) 37–51.
[2] M. Balabanovic, Y. Shoham, Content-based, collaborative recommendation, Commun. ACM 40 (3) (1997) 66–72.
[3] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowl.-Based Syst. 26 (1–2) (2012) 225238.
[4] J. Bobadilla, F. Ortega, A. Hernando, A. Gutirrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.
[5] J.S. Breese, D. Heckerman, C.M. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24–26, 1998, pp. 43–52.
[6] L. Candillier, F. Meyer, F. Fessant, Designing specific weighted similarity measures to improve collaborative filtering systems, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), Leipzig, Germany, July 16–18, 2008, pp. 242–255.
[7] L. Chen, R. Nayak, Y. Xu, Improving matching process in social network, in: Proceedings of the 10th IEEE International Conference on Data Mining Workshops (ICDMW 2010), Sydney, Australia, 2010, pp. 305–311.
[8] C.T. Chu, S.K. Kim, Y.A. Lin, Y.Y. Yu, G. Bradski, A.Y. Ng, K. Olukotun, Map-reduce for machine learning on multicore, in: Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4–7, 2006, pp. 281–288.
[9] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: Proceedings of the 2010 ACM Conference on Recommender Systems, (RecSys 2010), Barcelona, Spain, September 26–30, 2010, pp. 39–46.
[10] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure Appl. Math. 57 (11) (2004) 1413–1457.
[11] C. Desrosiers, G. Karypis, A novel approach to compute similarities and its application to item recommendation, in: Proceedings of 11th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2010), 2010, pp. 39–51.
[12] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, IEEE Trans. Knowl. Data Eng. 19 (3) (2007) 355–369.
[13] Z. Huang, H. Chen, D.D. Zeng, Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004) 116–142.
[14] T. Hofmann, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004) 89–115.
[15] M. Jamali, M. Ester, TrustWalker. A random walk model for combining trust-based and item-based recommendation, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28–July 1, 2009, pp. 397–406.
[16] R. Jin, J.Y. Chai, L. Si, An automatic weighting scheme for collaborative filtering, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04), Sheffield, UK, July 25–29, 2004, pp. 337–344.

[17] J.A. Konstan, J. Riedl, B. Smyth, Private distributed collaborative filtering using estimated concordance measures, in: Proceedings of the 2007 ACM Conference on Recommender Systems, (RecSys 2007), Minneapolis, MN, USA, October 19–20, 2007, pp. 1–8.
[18] Y. Koren, R. Bell, V. Chris, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[19] H.J. Kwon, K.S. Hong, Novel neighbor selection method to improve data sparsity problem in collaborative filtering, Int. J. Distrib. Sens. Netw. 2013 (2013). Article ID 847965.
[20] H. Liu, Z. Hu, A.U. Mian, H. Tian, X. Zhu, A new user similarity model to improve the accuracy of collaborative filtering, Knowl.-Based Syst. 56 (2014) 156–166.
[21] M. Zinkevich, A. Smola, J. Langford, Slow learners are fast, in: Proceedings of 23rd Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 2009, pp. 2331–2339.
[22] P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in: Recommender Systems Handbook, Springer, 2011, pp. 73–105.
[23] H. Luo, C. Niu, R. Shen, C. Ullrich, A collaborative filtering framework based on both local user similarity and global user similarity, Mach. Learn. 72 (3) (2008) 231–245.
[24] X. Luo, Y. Xia, Q. Zhu, Applying the learning rate adaptation to the matrix factorization based collaborative filtering, Knowl.-Based Syst. 37 (2013) 154–164.
[25] X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, Knowl.-Based Syst. 27 (2012) 271–280.
[26] H. Ma, I. King, M.R. Lyu, Effective missing data prediction for collaborative filtering, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07), Amsterdam, The Netherlands, July 23–27, 2007, pp. 39–46.
[27] B. ODonoghue, C. Emmanuel, Adaptive restart for accelerated gradient schemes, Found. Comput. Math. (2013) 1–18.
[28] P. Pirasteh, J.J. Jung, D. Hwang, Item-based collaborative filtering with attribute correlation: a case study on movie recommendation, in: Proceedings of 6th Asian Conference on Intelligent Information and Database Systems (ACIIDS (2)), Bangkok, Thailand, April 7–9, 2014, pp. 245–252.
[29] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the International Conference on Computer Supported Cooperative Work (CSCW), Chapel Hill, NC, USA, October 22–26, 1994, pp. 175–186.
[30] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: Recommender Systems Handbook, Springer, 2011, pp. 1–35.
[31] B.M. Sarwar, G. Karypis, J.A. Konstan, J.T. Riedl, Application of dimensionality reduction in recommender system-a case study, in: Proceedings of ACM WebKDD 2000 Web Mining for E-commerce Workshop, 2000.
[32] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02), Tampere, Finland, August 11–15, 2002, pp. 253–260.
[33] U. Shardanand, P. Maes, Social information filtering: algorithms for automating word of mouth, in: Proceedings of the 1995 International Conference on Computer–Human Interactions (CHI'95), Denver, Colorado, USA, May 7–11, 1995, pp. 210–217.
[34] A. Strehl, J. Ghosh, R. Mooney, Impact of similarity measures on web-page clustering, in: Proceedings of the International Workshop on Artificial Intelligence for Web Search, 2000, pp. 58–64.
[35] L. Xiao, T. Zhang, A proximal stochastic gradient method with progressive variance reduction, SIAM J. Optim. 24 (4) (2014) 2057–2075.
[36] C.P. Wei, R.F. Easley, M.J. Shaw, Web-based recommendation systems for personalized e-commerce shopping, in: E-Business Management, Springer, 2002, pp. 249–276.
[37] M. Weimer, A. Karatzoglou, A.J. Smola, Improving maximum margin matrix factorization, Mach. Learn. 72 (3) (2008) 263–276.

Please cite this article in press as: P. Pirasteh et al., Exploiting matrix factorization to asymmetric user similarities in recommendation systems, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.006