CHIN. PHYS. LETT. Vol. 34, No. 6 (2017) 068902

User Heterogeneity and Individualized Recommender *

Qing-Xian Wang(王庆先)[1], Jun-Jie Zhang(张君杰)[2], Xiao-Yu Shi(史晓雨)[2], Ming-Sheng Shang(尚明生)[2**]

[1]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054*

[2]*Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714*

(Received 3 March 2017)

Previous works on personalized recommendation mostly emphasize modeling peoples' diversity in potential favorites into a uniform recommender. However, these recommenders always ignore the heterogeneity of users at an individual level. In this study, we propose an individualized recommender that can satisfy every user with a customized parameter. Experimental results on four benchmark datasets demonstrate that the individualized recommender can significantly improve the accuracy of recommendation. The work highlights the importance of the user heterogeneity in recommender design.

PACS: *89.75.Hc, 05.10.−a, 89.70.Cf* DOI: 10.1088/0256-307X/34/6/068902

In big data era, recommender systems are considered as the most effective technology for solving the information overload problem and are applied in a variety of applications. They focus on predictions that fit the wishes and needs of users. Recommendation algorithm, as the key and core element of the recommender system, has been proposed based on various ideas and concepts[1,2] such as simple aggregates, search-based recommendations, or tailored to individual users,[3] e.g., the content-based method,[4] collaborative filtering algorithm (CF)[5,6] (including the matrix factorization method,[7]) network-based method,[8,9] and hybrid method.[10]

Regarding personalized recommendation, existing works mostly emphasize modeling peoples' diversity in potential favorites into a uniform recommender; even for the hybrid method that integrates two different algorithms with a weighted parameter, it can still be regarded as a uniform recommender. For example, Zhou *et al.*[10] proposed a hybrid method (HHP) combining heat conduction and probability spreading algorithm. With a fixed hybrid parameter, the HHP method provides a smooth yet non-trivial transition from one method to the other.

However, these uniform recommenders are not one-size-fits-all. Different recommenders have different virtues, making them suitable for different users. It has been proven that due to the fact that inactive users with low degree have not much experience in exploring new items, they are more likely to select popular items, while active users prefer to try novel items.[11] Hence, it is not possible to meet the needs of users via adopting the uniform recommender. Meanwhile, it is necessary and important to design an individualized recommender with consideration of user heterogeneity, which further diversifies the recommender to amplify diversity among different individuals.[12]

In such context, a few works have been proposed to address the aforementioned issue. The individualized recommender can be implemented in two different ways including parameter-level individualization and algorithm-level individualization. For the algorithm-level individualization, to take advantage of the relative merits of different algorithms, Ekstrand *et al.*[13] designed a new version of the MovieLens movie recommender that supports a multiple recommender and allows users to choose the recommender they intend to provide their recommendations. With the consideration of heterogeneity of a real user, Shi *et al.*[14] proposed a novel personalized recommender based on user preferences, which allows multiple recommenders to exist in the recommender system simultaneously. For the parameter-level individualization, Guan *et al.*[15] proposed a user-oriented hybrid algorithm (UHHP) based on the HHP method. With a tunable hybrid parameter, UHHP allows each user to have his/her own individualized hybrid parameter to recommendation. The recommendation performance of UHHP can be improved to some extent, but it has studied mainly on the parameter-level individualization of uniform recommender with the hybrid network-based method.

In this Letter, we focus on the parameter-level individualization of uniform recommender with a single algorithm. We adopt the slope one algorithm, a kind of important and classic CF, as the original algorithm. In the original slope one algorithm with different schemes, the slope parameter α is fixed (i.e, $\alpha = 1$). To achieve the parameter-level individualization of the slope one algorithm, we first design a tunable slope parameter that allows each user to have an individualized slope parameter $\alpha_{\rm u}$, which is adjustable. Interestingly, we find that the recommendation performance is varying with different α, and different users have the different optimal $\alpha_{\rm u}$. With

*Supported by the National Natural Science Foundation of China under Grant Nos 91646114, 61602434 and 61370150, and the Youth Innovation Promotion Association of Chinese Academy of Sciences under Grant No 2017393.

**Corresponding author. Email: msshang@cigit.ac.cn

this understanding, we propose an individualized slope α recommender (ISα) based on the bi-polar slope one scheme. Furthermore, there is not an effective method for determining the optimal individual parameter in the real world, but the optimal value exists.[16] We design a feasible method to calculate the suitable α for each user (i.e., $\alpha_{\rm u}$) in this work. In such a setting, we find that the prediction accuracy of ISα outperforms the original bi-pole slope one scheme.

To begin our analysis, a CF-based recommender system generally models historical user behaviors into a user-item rating-matrix and the problem of CF-based recommendation is described as follows: given the item set I and the user set U, the user–item rating matrix R is as $|U|\times|I|$ matrix where each element $r_{u,i}$ is connected to user u preference on item i. Then let $R_{\rm train}$ and $R_{\rm test}$ denote the known and unknown entry sets in R, respectively, given as the training and test dataset. The problem of CF is how to construct an estimator $\hat{R}$, which can achieve or approximate

$$\arg\min(\sum_{(u,i)\in R_{\rm test}} |\hat{r}_{u,i} - r_{u,i}|). \tag{1}$$

The original slope one algorithm is one of the rating-based CF recommenders.[17] Formally, the slope one algorithm is in the form

$$f(x) = x + b, \tag{2}$$

where b is a constant, and x is a variable representing the rating values. It subtracts the average ratings of two items to measure how much more, on average, one item is more liked than another. This difference is used to predict another user's rating of one of these two items, given his rating of the other.

For the basic slope one scheme (SO), the prediction rating of item j by user u is

$$r^{\rm so}_{u,j} = \bar{r} + \frac{1}{|R(u,i)|}\sum_{j\in R(u,i)} {\rm dev}_{i,j}, \tag{3}$$

where $\bar{r} = \frac{\sum_{j\in R(u,i)} r_{u,j}}{|R(u,i)|}$, $R(u,i)$ is the set of items that have been both rated by u and co-rated with item i. The average deviation of item j with respect to item i is defined as

$${\rm dev}_{j,i} = \frac{\sum_{u\in S(i,j)} r_{u,j} - r_{u,i}}{|S(i,j)|}, \tag{4}$$

where $S(i,j)$ is the set of users who rate items i and j.

In the slope one algorithm, each user has the same slope parameter (i.e., 1), which means that everyone has the same scoring criteria. However, this assumption is not true in a real case since some pessimistic users prefer giving a lower score to all items, even the liked items, while the optimistic users may give a higher score to all items, including some items they dislike. Therefore, for supporting all types of users including balanced, optimistic, pessimistic, and bimodal users, we apply a tunable slope parameter to Eq. (2). Specifically, we change the linear regression form in Eq. (2) to the slope-α form

$$f(x) = \alpha x + b, \tag{5}$$

where the slope parameter α is a tunable variable, instead of $\alpha = 1$ for all users. Taking SO as an example, the slope α reads

$$r^{\rm so}_{u,i} = \alpha\bar{r} + \frac{1}{|R(u,i)|}\sum_{j\in R(u,i)} {\rm dev}_{i,j}. \tag{6}$$

In this study, we conduct experiments on four datasets: FilmTrust, MovieLens, Jester and EachMovie. In these datasets, a higher rating indicates that the user likes the item more. To obtain an objective result, we normalize all of the rating data into the range of $[0, 5]$. The descriptions of these four datasets are listed in Table 1, where $|E|$ is the number of ratings in the dataset. The accuracy of recommendation is one of the most important evaluation metrics for recommenders. Mean absolute error (MAE) is the most popular metrics used in evaluating accuracy of CF models. Formally, the MAE of a recommender is given by

$$MAE = \frac{\sum_{u,i\in R_{\rm test}} |r_{u,i} - \hat{r}_{u,i}|}{|R_{\rm test}|}. \tag{7}$$

Table 1. Properties of the four datasets.

Dataset	$\|U\|$	$\|I\|$	$\|E\|$	Sparsity
FilmTrust	1508	2071	35497	1.14×10^{-2}
ML1M	6040	3952	1000209	4.19×10^{-2}
Jester	50692	140	1728847	2.436×10^{-1}
EachMovie	61265	1623	2811718	2.83×10^{-2}

For investigating the user heterogeneity in the slope-α algorithm, we change the value of α in the range of $[0.9, 1.1]$ to calculate the predicted ratings by using Eq. (6), where the step length of α is 0.005. The range of $[0.9, 1.1]$ is an empirical value, which is a trade-off between the experiment complexity and computational cost. Note that the range of $[0.9, 1.1]$ is determined by the idea of proximity search and the original slope value is 1. Then we compute the MAE of each user, and choose the minimal value of MAE as the best α for each user. Figure 1 describes the distributions of α in terms of MAE. For example, there are two peaks in FilmTrust. The larger one is close to $\alpha = 0.9$ (i.e., around 22% users with this optimal α), while the smaller one is close to 1.10 (i.e., around 12% users with this optimal α). The same phenomenon also happens in Jester. In ML1M and EachMovie, there is an obvious peak in $\alpha = 0.9$ and other values are uniformly distributed between $[0.9, 1.1]$. These results indicate that users have quite different individualized slope parameters $\alpha_{\rm u}$ in real systems. If we use the same slope parameter for all users as the original slope one algorithm, many users cannot receive the best recommendation.

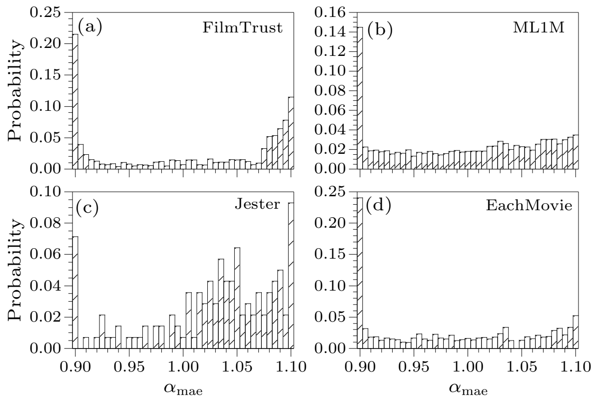

Fig. 1. The distribution of α in each dataset: (a) FilmTrust, (b) ML1M, (c) Jester and (d) EachMovie.

According to the above analysis, we propose an individual slope α recommender (ISα) based on the bipolar slope one scheme (BPSO), since BPSO is an important improvement of the slope one algorithm and is widely used in real applications. We first introduce BPSO, which is based on dividing the set of all items into items liked and disliked by a given user. A common way to identify liked and disliked items is to apply the user's average rating as a threshold. In such a setting, the liked set of items is that the users' ratings are higher than the threshold awarded by the given user, while the disliked set of items includes those where the users' ratings are lower than this threshold. From these liked and disliked items, two separate predictions are derived, which are combined into one prediction finally. The sets of users who like and dislike are denoted by $S^{+1}(i,j)$ and $S^{-1}(i,j)$, respectively, with both i and j. The deviations for liked and disliked items are

$$\mathrm{dev}_{j,i}^{+1} = \frac{1}{|S^{+1}(j,i)|} \sum_{u \in S^{+1}(j,i)} (r_{u,j} - r_{u,i}), \quad (8)$$

$$\mathrm{dev}_{j,i}^{-1} = \frac{1}{|S^{-1}(j,i)|} \sum_{u \in S^{-1}(j,i)} (r_{u,j} - r_{u,i}). \quad (9)$$

Then the prediction for the rating of item j based on the rating of item i is either $r_{u,i} + \mathrm{dev}_{j,i}^{+1}$ or $r_{u,i} + \mathrm{dev}_{j,i}^{-1}$ depending on whether the target user u likes or dislikes item i, respectively.

The BPSO is thus given by

$$\hat{r}_{u,j}^{bi} = \frac{\sum_i |S^{+1}(j,i)| P_{j,i}^{+1} + \sum_i |S^{-1}(j,i)| P_{j,i}^{-1}}{\sum_i |S^{+1}(j,i)| + \sum_i |S^{-1}(j,i)|}, \quad (10)$$

where $P_{j,i}^{+1}$ is the prediction score of item j in liked set (i.e., $P_{j,i}^{+1} = (r_{u,i} + \mathrm{dev}_{j,i}^{+1})$), $P_{j,i}^{-1}$ is the prediction score of item j in disliked set (i.e., $P_{j,i}^{-1} = (r_{u,i} + \mathrm{dev}_{j,i}^{-1})$).

For applying the slope parameter to an individual level, we allow each user to adjust his/her individualized parameter $\alpha_{\rm u}$ to obtain the best recommendation. Hence, the ISα consists of the following steps: firstly, we divide the set of all items into items liked and disliked by applying the target user u's average rating as threshold. Then the deviations of liked and disliked subsets are calculated according to Eq. (8). After that, in each subset the predicted value on item j based on item i can be written as $\alpha_u r_{u,i} + \mathrm{dev}_{j,i}^{+}$ and $\alpha_u r_{u,i} + \mathrm{dev}_{j,i}^{-}$, respectively. Finally, the predicted rating $\hat{r}_{u,j}$ is given by

$$\hat{r}_{u,j}^{IS\alpha} = \frac{\sum_i |S^{+1}(j,i)| P_{j,i}^{per^{+1}} + \sum_i |S^{-1}(j,i)| P_{j,i}^{per^{-1}}}{\sum_i |S^{+1}(j,i)| + \sum_i |S^{-1}(j,i)|}, \quad (11)$$

where $P_{j,i}^{per^{+1}}$ is the prediction score of item j in liked set (i.e., $P_{j,i}^{per^{+1}} = (\alpha_u r_{u,i} + \mathrm{dev}_{j,i}^{+1})$), $P_{j,i}^{per^{-1}}$ is the prediction score of item j in disliked set (i.e., $P_{j,i}^{per^{-1}} = (\alpha_u r_{u,i} + \mathrm{dev}_{j,i}^{-1})$).

However, it is difficult to find the best individualized parameter $\alpha_{\rm u}$ for each user in the real world. Hence, we adopt a simple binary classification method to determine the suitable $\alpha_{\rm u}$ for each user. In detail, we first compute the global average rating of all users (i.e., $\bar{r}$) and average rating of the target user u (i.e, $\bar{r}_u$). Then we use the global rating as the threshold. If the average rating of the target user is higher than the global average rating, then we will set $\alpha_{\rm u}$ beyond 1. If not, we will set $\alpha_{\rm u}$ below 1. It can be described by

$$\alpha_{\rm u} = \begin{cases} 1+\theta, & \bar{r}_{\rm u} \geq \bar{r}, \\ 1-\theta, & \bar{r}_{\rm u} < \bar{r}, \end{cases} \quad (12)$$

where θ is a tunable variable in the range of $[0, 0.1]$.

For the computation complexity of ISα, the cost of calculating the prediction rating for each user is $O(|U||I|^2)$, and the cost of finding the optimal θ is related to the defined search space $[\theta_{\min}, \theta_{\max}]$ and step length. Thus the computation complexity of ISα is just a small multiple of the complexity compared with BPSO.

For analyzing the suitable $\alpha_{\rm u}$ with θ of ISα, we change the value of θ in the range of $[0, 0.1]$ to calculate the predicted ratings. Note that the optimal $\alpha_{\rm u}$ for each user exists theoretically. Here we try to find an approximate $\alpha_{\rm u}$ with θ by using Eq. (12). The step length of θ is 0.005. Figure 2 shows the MAE of prediction with different θ in ISα. Here $\theta = 0$ corresponds to the case of original BPSO, while the other values represent the ISα with different $\alpha_{\rm u}$. For FilmTrust, the trend of MAE decreases sharply when θ increases from 0 to 0.080, and the minimal MAE is arrived at $\theta = 0.080$ (i.e., $\alpha_{\rm u} = 0.920$ and 1.080 in Eq. (11)). After that, the value of MAE increases slowly when the value of θ is in the range of $[0.080, 0.10]$. The same phenomena also happen in the other three datasets. The optimal θ values of ML1M, Jester and EachMovie are 0.045, 0.060 and 0.080, respectively. Thus the results show that the performance of recommender varies with θ and the optimal values of θ for different datasets are different. On the other hand, the stability of individual parameters directly determines the effectiveness of ISα. From Fig. 2 we can see that the optimal tunable parameter $\theta_{\rm opt}$ can be found in the range of $[0, 0.1]$, although the value of $\theta_{\rm opt}$ is varying in different datasets. Hence, it also demonstrates that the range of individual parameter is stable.

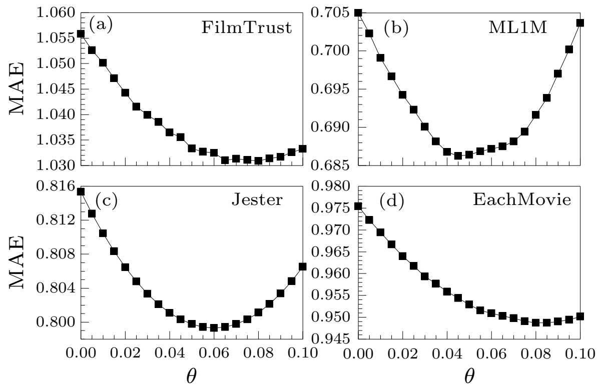

Fig. 2. Performance of the ISα recommender with different θ in terms of MAE: (a) FilmTrust, (b) ML1M, (c) Jester and (d) EachMovie.

Compared with the original BPSO, we use the ISα to recommendation, where ISα uses the optimal θ in Eq. (12) to recommendation, as shown in Fig. 2. Table 2 lists the result of the prediction accuracy, which indicates that the recommendation accuracy can be improved to some extent when using ISα. Thus it proves that our proposed ISα is feasible and effective.

Table 2. The MAE comparison between ISα and original BPSO in four datasets.

Algorithm	FilmTrust	ML1M	Jester	EachMovie
BPSO	1.0558	0.7050	0.8153	0.9754
ISα	1.0309	0.6862	0.7993	0.9487
Enhancement	2.36%	2.67%	1.96%	2.74%

In summary, we have studied the issue of user heterogeneity in rating-based prediction. For the slope one algorithm, we find that real users are quite different in their optimal individualized slope parameters. We propose the ISα algorithm to recommendation by applying the original BPSO algorithm to individual level, thus each user has an individualized slope parameter $\alpha_{\rm u}$ to adjust. Moreover, we test the ISα in four benchmark datasets and find that our method can further improve the prediction accuracy in terms of MAE compared with the original BPSO algorithm.

To accurately estimate the optimal tunable parameter in the recommender system for future recommendation is still a challenge. So far, the usual way to solve this problem is based on the analysis of the history data. Normally, the history data is divided into training set and probe set. The parameter for future recommendation is determined when the algorithm achieves its best performance in this training probe set division. In our work, we also adopt this way to estimate our optimal parameter. However, this method is not the best one. For example, a user's optimal parameter will change with time in real systems. Therefore, analyzing a user's history of activity records with time information may lead to a deeper understanding of the user's behavior pattern and thus a better prediction of their individualized parameters. This problem will be a part of our future work. Furthermore, the computational efficiency will no longer be a major problem for ISα, according to the development of computing technologies.

References

[1] Ekstrand M D, Riedl J T and Konstan J A 2011 *Found. Trends Human-Comput. Interaction* **4** 175
[2] Lü L, Medo M, Yeung C H, Zhang Y C, Zhang Z K and Zhou T 2012 *Phys. Rep.* **519** 1
[3] Pan X, Deng G S and Liu J G 2010 *Chin. Phys. Lett.* **27** 068903
[4] Van den Oord A, Dieleman S and Schrauwen B 2013 *Adv. Neural Inf. Process. Syst.* (Lake Tahoe Nevada USA 5–10 December 2013) p 2643
[5] Resnick P, Iacovou N, Suchak M, Bergstrom P and Riedl J 1994 *Proc. ACM Conf. Comput. Supported Cooperative Work* (Chapel Hill NC USA 22–26 October 1994) p 175
[6] Linden G, Smith B and York J 2003 *IEEE Internet Computing* **7** 76
[7] Bell R, Koren Y and Volinsky C *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Jose California, USA 12–15 August 2007) p 95
[8] Deng S, Huang L and Xu G 2014 *Expert Syst. Appl.* **41** 8075
[9] Shang M S and Zhang Z K 2009 *Chin. Phys. Lett.* **26** 118903
[10] Zhou T, Kuscsik Z, Liu J G, Medo M, Wakeling J R and Zhang Y C 2010 *Proc. Natl. Acad. Sci. USA* **107** 4511
[11] Shang M S, Lü L, Zhang Y C and Zhou T 2010 *Europhys. Lett.* **90** 48006
[12] Chen L, de Gemmis M, Felfernig A, Lops P, Ricci F and Semeraro G 2013 *ACM Trans. Interactive Intelligent Syst.* **3** 17
[13] Ekstrand M D, Kluver D, Harper F M and Konstan J A *Proceedings of the 9th ACM Conference on Recommender Systems* (Vienna, Austria 16–20 September 2015) p 11
[14] Shi X Y, Shang M S, Luo X, Khushnood A and Li J 2017 *Physica* A **467** 490
[15] Guan Y, Zhao D, Zeng A and Shang M S 2013 *Physica* A **392** 3417
[16] Shang M S 2016 *J. Chin. West Normal University* **1** 61 (in Chinese)
[17] Lemire D and Maclachlan 2005 *ASDM: Proceedings of the 2005 SIAM International Conference on Data Mining* p 471