Li *et al. Hum. Cent. Comput. Inf. Sci.* (2015) 5:21
DOI 10.1186/s13673-015-0041-2

Human-centric Computing and Information Sciences
a SpringerOpen Journal

RESEARCH

Open Access

Personalized fitting recommendation based on support vector regression

Weimin Li[1,2], Xunfeng Li[1], Mengke Yao[1], Jiulei Jiang[3] and Qun Jin[4*]

*Correspondence:
jin@waseda.jp
[4] Faculty of Human Sciences, Waseda University, Tokorozawa, Japan
Full list of author information is available at the end of the article

Abstract

Collaborative filtering (CF) is a popular method for the personalized recommendation. Almost all of the existing CF methods rely only on the rating data while ignoring some important implicit information in non-rating properties for users and items, which has a significant impact on the preference. In this study, considering that the average rating of users and items has a certain stability, we firstly propose a personalized fitting pattern to predict missing ratings based on the similarity score set, which combines both the user-based and item-based CF. In order to further reduce the prediction error, we use the non-rating attributes, such as a user's age, gender and occupation, and an item's release date and price. Moreover, we present the deviation adjustment method based on the support vector regression. Experimental results on MovieLens dataset show that our proposed algorithms can increase the accuracy of recommendation versus the traditional CF.

Keywords: Collaborative filtering, Recommendation, Personalized fitting, Deviation adjustment

Background

With the rapidly growing information available on the Internet, people have to spend much more time selecting useful information. To solve the information overload problem, recommender systems have emerged. In recent years, recommender systems have widely used in e-commerce and social network to supply users with personalized information [1]. Collaborative filtering is one of the most successful techniques for its simplicity and efficiency, and it is a good complementary technology to the content-based filtering [2–7]. Its key process is to find similar users for the target user, or similar items for the predicted item. However, there still exist some inherent problems to be addressed and solved, such as accuracy, data sparsity, cold start and scalability.

There is an important phenomenon that the average rating of users and items has shown a certain stability in a certain time period, which helps us predict the score become possibility. In order to improve the quality of recommendation, various improved approaches, such as Singular value decomposition (SVD) [8], Biparite network [9, 10] and Random walk [11], were introduced to collaborative filtering.

However, all these methods ignore many latent users and items features. For example, in the MovieLens datasets, we can easily find that most students would prefer to the fantasy movies, and the popularity of comedy movies far surpasses drama. Under this condition, rating data is not functional to recommend a suitable film to users. These approaches, which are almost the same as the traditional similarity-based collaborative filtering, only consider the rating data. Instead, a few important issues are not involved in the above reference. In this study, we firstly find the similarity set of users and items. Most methods consider the user or item independently. We construct the user similarity set and item similarity set, combine them and model the linear relation between the total trusty set and precision results. Especially, all methods are improved to give better recommendation, but don't take the errors of forecasting into consideration. In this paper, we find that there are some relations between the errors of forecasting and the features of users and items, and try to establish an error feedback mechanism to improve the recommendation.

Considering that the user and item information may be a key factor in the appropriate recommendation, we use not only the average rating, but also the non-rating attributes [12], such as a user's age, gender and occupation, and an item's release date and price. By using this information, we can model a dynamic deviation adjustment based on support vector regression (SVR) [13]. The objective is to find the relation between the non-rating attributes and precision errors. Then we can adjust the precision errors and improve the recommendation effectively by making use of the information of users' and items'characteristic [14].

In this paper, we make efforts to exploit a new score prediction method. We design a personalized fitting pattern by using a training set which comes from the similarity score set with regard to the target user and target item. Particularly, we use the non-rating features which include both the user's and the item's features to further lower the residual error using SVR. The related experiments show that our proposed approach is more effective than both the traditional user-based CF and item-based CF.

The rest of this paper is organized as follows. "Related work" discusses the related work of Collaborative filtering and support vector machine (SVM). In "Personalized fitting", we present our design of a personalized fitting pattern by using a training set, and we use the non-rating features. Deviation adjustment by SVR is described in "Deviation adjustment by support vector regression". We verify our method using the MovieLens datasets, and some comparisons are discussed in "Experiments". Finally, we make conclusions and outlook the future work in "Conclusion".

Related work

Collaborative filtering (CF) is the most mature and popular method in recommender systems for its effective and simplification. There are many CF-based recommendation systems developed in the academy and industry that are often based on the assumption that the target user will prefer the items with the similar preferences of other users. Collaborative filtering can be divided into two categories: Memory-based CF and Model-based CF. In Memory-based CF, recommended items are those that were preferred by the users who share the similar preferences as the target user (User-based CF), or those that are similar to the other items preferred by the target user (Item-based CF). They are also called similarity-based CF since finding the most similar users or items is of great importance. Commonly used similarity measures include consine similarity [15], adjust Cosine similarity and Pearson correlation coefficient [16].

Up to now, many improved similarity approaches have been proposed to improve the quality of the recommendation. Breese [17] found that common behavior on less popular items can better reflect users' preferences, and proposed interests partition function to adjust the similarity method. To overcome the data sparse problem in collaborative filtering, structural similarity between users [18] and objects based on bipartite network model [10] were proposed.

Model-based collaborative filtering including probabilistic model [19], Bayesian Model [20], factorization model [21] and latent class models [22] use the ratings data to perform training. SVD uses the low level rating matrix to predict the true score matrix. Memory-based CF is very simple and easy to implement, and model-based CF shows the great advantage in expansibility and flexibility. Most of all, the recommendation systems in E-commerce, such as Amazon books and Ebay shopping recommender system, are using traditional collaborative filtering methods.

There are many efforts to make in order to improve the personalized recommendation system [23, 24]. Tagging is an important information for personalized recommendation. Zhang et al. proposed an integrated diffusion-based algorithm by making use of both the user-item relations and the collaborative tagging information [25]. Shepitsen et al. presented a personalization algorithm for recommendation in folksonomies, which relies on hierarchical tag clusters [26]. Song et al. modeled a user's adoption pattern as an information flow network for a recommendation system. The authors proposed an early adoption based information flow (EABIF) network by comparing the timestamps when users access documents, and a topic-sensitive early adoption based information propagation (TEABIF) network according to the topics of the documents users accessed [27].

Though there are many methods to improve the accuracy of recommendation, all methods did not use the similarity set of users and items for personalized recommendation. Furthermore, the prediction errors are inevitable and deviation adjustment is not involved to improve accuracy in these methods.

Personalized fitting

In this section, we define the similarity score set, and then outline the Personalized Fitting (PF) framework.

Let $U = \{u_1, u_2, u_3, \ldots, u_M\}$ be the set of users, and $S = \{s_1, s_2, s_3, \ldots, s_N\}$ be the set of items in the recommender system. We assume $r_{m,n}$ is the rating given to item $s_n \in S$ by the user $u_m \in U$. The history scoring record is presented with a $M \times N$ matrix as shown in Table 1. All those ratings in Table 1 represent the user historical behaviors.

Similarity measurement

There are many methods used to compute the similarity between the users and items in the collaborative recommender systems. The most popular methods are Cosin-based and Person correlation coefficient, which are all based on the rating matrix. In Cosine-based method, two users u_m and u_n are treated as two rating vectors that both reviewed. Therefore, the similarity is equal to the two vectors' cosine power as follows.

$$sim(u_m, u_n) = cos\left(\overrightarrow{u_m}, \overrightarrow{u_n}\right) = \frac{\sum_{s_i \in C_{m,n}} r_{m,i} r_{n,i}}{\sqrt{\sum_{s_i \in C_{m,n}} r_{m,i}^2} \sqrt{\sum_{s_i \in C_{m,n}} r_{n,i}^2}} \tag{1}$$

Table 1 A history record matrix example

	s_1	s_2	s_3	…	s_4	…	s_5
u_1	1	2	4	…	4	…	1
u_2		3	3	…		…	
u_3	5	1		…	3	…	4
…	…	…	…	…	…	…	…
u_m	2	4	1	…	$r_{m,n}$	…	5
…	…	…	…	…	…	…	…
u_m	2		3	…	5	…	2

The rating similarity can also be measured by Person correlation coefficient, which can be given as follows.

$$sim(u_m, u_n) = \frac{\left|\sum_{s_i \in R_m \cap R_n} \left(r_{m,i} - \overline{u_m}\right)\left(r_{n,i} - \overline{u_n}\right)\right|}{\sqrt{\sum_{s_i \in R_m \cap R_n} \left(r_{m,i} - \overline{u_m}\right)^2} \sqrt{\sum_{s_i \in R_m \cap R_n} \left(r_{n,i} - \overline{u_n}\right)^2}} \tag{2}$$

where $R_m(R_n)$ is the set of records rated by $u_m(u_n)$. The correlation $\mathrm{corr}(u_m, u_n)$ between u_m and u_n is computed on the records $C_{m,n} = R_m \cap R_n$ rated by both u_m and u_n, and $\overline{u_m}$ and $\overline{\mathrm{u}_n}$ indicates the average scores of u_m and u_n on all records of R_m and R_n, respectively. We can also calculate the similarity between items s_m and s_n using the same principle as the user's similarity. All the similarity degree in this paper uses this Pearson Correlation Coefficient.

Similarily score set

In similarity-based collaborative recommender systems, k-nearest users or items must be chosen as their trustworthy neighbors firstly. Similar to the similarity-based collaborative filtering method, our main purpose is also to predict the rating $\hat{r}_{m,n}$ of the target user $u_m \in U$ for the target item $s_n \in S$ that he/she has not known yet. But our predicting method is based on the Similarity Score Set. Therefore, we should firstly find similar user set of the target user and similar item set of the target item, and then base on these definitions Similarity Score (SS) Set will be illustrated. Here we give these three definitions.

Definition 1 Similar Users Set (SU) of a target user u_m is k users that are the most similar with target user u_m, which suggests that similar user's influence on his/her behavior is reliable.

$$\mathrm{SU}(u_m, k) = \{u_i \in U | sim(u_m, u_i)\text{ranked in the top k with the user similartiy}\} \tag{3}$$

Definition 2 Similar items set (SI) of a target item s_i. Is k items that are the most similar to target item s_n. That is to say those items may share the similar popularity.

$$SI(s_m, k) = \{s_j \in S | sim\left(s_n, s_j\right)\text{ranked in the top k with the item similartiy}\} \tag{4}$$

Definition 3 Similarity score set (SS) of a target user u_m and a target item s_n are those rating records which belong to the existing ratings of similar users set (SU) to the similar items set (SI).

$$SS(u_m, s_n, k_u, k_s) = \left\{ \left(u_i, s_j\right) | u_i SU(u_m, k_u) \cap s_j \in SI(s_n, k_s) \cap r_{i,j} \neq \emptyset \right\} \tag{5}$$

where k_u and k_s are the similar user size and similar item size regulation parameters, respectively. $\mathrm{SU}(u_m, k_u)$ is the k_u nearest neighborhoods users of u_m, and $\mathrm{SI}(s_n, k_s)$ is the k_s nearest neighborhoods items of s_n. In this paper, in order to improve the precision of forecasting, we proceed a further optimization on training data and construct the similarity score set. The Similarity Score Set is a reliable training set in predicting rating since it takes the advantages of both the user-based and item-based collaborative filtering approaches.

Personalized fitting

To predict rating $\hat{r}_{m,n}$, there exists a linear relationship between the rating $r_{m,n}$ and the average rating of both u_m and s_n. The Similarity Score Set $SS(u_m, s_n, k_u, k_s)$ can be easily obtained when the adjusting parameters k_u and k_s are given. Here we suppose a tuple $(u_i, s_j) \in SS(u_m, s_n, k_u, k_s)$, then we define the Personalized Fitting triples as $\delta_k\left(\overline{u_i}, \overline{s_j}, r_{i,j}\right)$, where $\overline{u_j} = (\frac{1}{|R_i|}) \sum_{s_j \in R_i} r_{i,j}$ and $R_i = \{s_j \in S | r_{i,j} \neq \emptyset\}$;$\overline{S_j} = \left(\frac{1}{|T_j|}\right) \sum_{u_i \in R_j} r_{i,j}$and $T_j = \left\{u_j \in U | r_{i,j} \neq \emptyset\right\}$ and $0 < k \leq \left|SS(u_m, s_n, k_u, k_s)\right|$. In order to simplify the later descriptions, we generalize the Personalized Fitting triples as $\delta_k\left(x_k, y_k, z_k\right)$. x_k, y_k, z_k satisfy Eq. (6), and to get the best result is equal to adjust the parameters λ_m and μ_n. Loss function describes the proximity between the predicted value and the true value under different parameter. The adjusting parameters λ_m and μ_n can be obtained by means of minimizing the following loss function.

$$Los(\lambda_m, \mu_n) = \sum_k \left(\lambda_m x_k + \mu_n y_k - z_k\right)^2 \tag{6}$$

In general, the least square and gradient descent [28] can be used to minimize Eq. (6), but the gradient descent method can get higher precision in the shortest time.

$$\begin{cases} \frac{\partial Los}{\partial \lambda_m} = 2 \sum \left(\lambda_m x_k + \mu_n y_k - z_k\right) x_k \\ \frac{\partial Los}{\partial \mu_n} = 2 \sum \left(\lambda_m x_k + \mu_n y_k - z_k\right) y_k \end{cases} \tag{7}$$

In this paper, firstly, we take the derivatives with respect to parameters λ_m and μ_n. Then, according to the gradient descent method, we should update the parameters along the gradient descent direction. Therefore, the recursion formulas can be given as follows.

$$\begin{cases} \lambda_m = \lambda_m - \theta \frac{\partial Los}{\partial \lambda_m} \\ \mu_n = \mu_n - \theta \frac{\partial Los}{\partial \mu_m} \end{cases} \tag{8}$$

where the learning rate θ is assigned 0.001 generally, and parameters λ_m and μ_n are obtained by the gradient descent method as shown in Algorithm 1. In this study, predicting the rating of the target user u_m to the item s_n can be expressed as Eq. (9).

$$\hat{r}_{m,n} = \lambda_m \overline{u_m} + \mu_n \overline{s_n} \tag{9}$$

Algorithm 1. Gradient Descent

```
In: $R_m, TI$, a threshold $e$ (close to 0), learning step $\theta$,
Out: parameters $\lambda_m$ and $\mu_n$
1. Randomly initialized $\lambda_m$ and $\mu_n$
2. Compute$\frac{\partial Los}{\partial \lambda_m}$ and$\frac{\partial Los}{\partial \mu_m}$
3. Update $\lambda_m$ and $\mu_n$:
       $\lambda_m = \lambda_m - \theta \frac{\partial Los}{\partial \lambda_m}$
       $\mu_n = \mu_n - \theta \frac{\partial Los}{\partial \mu_m}$
4.  If($Los(\lambda_m, \mu_n) < e$)
        Return $\lambda_{m,} \mu_n$
    Else
        Goto 2
```

Personalized Fitting (PF) algorithm (shown in Algorithm 2) not only considers both user-based and item-based collaborative filtering, but also utilizes the average of user's rating and item's rating since they are important indicators of preferences in recommender systems .

Algorithm 2. Personalized Fitting (PF)

```
In:  User-item rating matrix ,
     user $u_m$ and item $s_n$, parameters $k_u$ and $k_s$
Out: predict rating $\hat{r}_{m,n}$
1. Compute the similarity between $u_m$ and all other users.
2. Add user $u_t(1 * t * M)$ who is ranked top $k_u$ with $u_m$ in user similarity.
3. Compute the similarity between $s_n$ and all other items.
4. Add item $s_t(1 * t * N)$ which is ranked top $k_s$ with $s_n$in item similarity.
5. Obtain fitting datasets$\delta_k\left(\overline{u_\imath}, \overline{s_\jmath}, r_{i,j}\right)$ according to $TU(u_m, k_u)$ and $TI(s_n, k_s)$
6. Compute fitting parameters $\lambda_m$ and $\mu_n$ according to Algorithm 1
7. Return $\hat{r}_{m,n} = \lambda_m \overline{u_m} + \mu_n \overline{s_n}$
```

Deviation adjustment by support vector regression

In the previous section, we considered that the rating $r_{m,n}$ only depends on the average rating $\bar{u}_m$ and $\bar{S}_n$. When we use the linear fitting model to describe their relationship, the predicted rating $\hat{r}_{m,n}$ can be obtained by the traditional CF or our proposed PF (Experiments using Personalized Fitting have shown better accuracy compared with the traditional similarity-based CF in "Experiments"). However, some other non-rating factors (such as a user's age, gender, occupation and an item's category, brand, etc.) are also important to affect $\hat{r}_{m,n}$. We assume a certain relationship exists between the residual $(r_{m,n} - \hat{r}_{m,n})$ and those non-rating factors. We propose a deviation adjustment method based on SVM to further improve the rating prediction accuracy.

SVR

Support Vector Machine was proposed by Vapnik [29, 30]. It is a universal machine learning algorithm based on solid statistical theory foundation. SVM learning algorithms are based on the structural risk minimization, which is different from the empirical error minimization used in the traditional machine learning algorithms. What's more, SVM has shown its great advantage in small sample learning, nonlinear classification and poor generalization ability [31]. Moreover, Vapnik expanded SVM to regression forecasting by adding ε-insensitive loss function, and built the SVR theory [32, 33]. The essence of SVR is the convex quadratic optimization problem. Its discriminant function is given as follows.

$$f(x) = \sum_{i=1}^{t} (a_i^* - a_i) K(x, x_i) + b \tag{10}$$

where $K(x, x_i)$ is the kernel function. Selecting the kernel function is the core step of SVR in solving the nonlinear regression problems. The basic idea is to transfer the original space into a new space $\Phi(x)$ by the kernel function. The new space is linearly separable. We only need to ensure the function like $K(u, v) = < \Phi(u), \Phi(v) >$ because it only uses the dot product in our new training model.

Deviation adjustment model

In this model, the residual $(r_{m,n} - \hat{r}_{m,n})$ comes from the user features P_m and the item features Q_n. Here, we selected the user's feature to include: gender (P_u^1), age (P_u^2), and occupation (P_u^3), and the item's feature to include: released year (Q_s^1) and genre (Q_s^2). Table 2 illustrates an example.

A stable SVR model can handle the user features and the item features to a deviation. We use the deviation to adjust the predicted rating to gain better results.

Experiments

Data set and setup

Our experiments were performed on a real and classical movie dataset MovieLens (http://www.movielens.umn.edu), which were collected by the GroupLens Research Project of Minnesota University. The main dataset includes 100,000 ratings from 943 users who reviewed 1,682 movies. The dataset also contains a script program which has split it into two parts: training set (80%) and testing set (20%). The biggest advantage of MovieLens dataset is that we can easily extract non-rating features for our work. In our experiments, we have extracted important user features which contain gender, occupation and age, the movie features which contain categories and release year.

Table 2 A simple example training data for our model

u_m	s_n	Deviation	P_u^1	P_u^2	P_u^3	Q_s^1	Q_s^2
1	15	0.35256	0	1	20	2	9
45	157	−0.24610	0	1	15	3	6
108	50	0.54350	1	2	4	1	4
204	123	1.23811	0	3	11	0	13
335	1001	−1.47634	1	1	8	2	4
….	….	….	…	…	…	…	…

In our work, we merely consider the precision as the only evaluation criterion to compare our method with the user-based CF and item-based CF. Mean absolute error (MAE) and Root mean squared error (RMSE) are the most widely used indicators in collaborative filtering. MAE and RMSE are defined as follows.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} \left| r_{m,n} - \hat{r}_{m,n} \right|}{N} \tag{11}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left| r_{m,n} - \hat{r}_{m,n} \right|^2}{N}} \tag{12}$$

where $r_{m,n}$ is the actual rating that user u_m gave to item s_n in the testing set, while $\hat{r}_{m,n}$ is the corresponding prediction rating calculated by certain methods using the training set. N stands for the number of testing records. Therefore, the smaller MAE and RMSE are, the better prediction quality of related method is.

Experimental result and analysis

In this paper, in order to validate the effectiveness of our PF algorithm, we compare our PF method with the traditional collaborative filtering methods, including the user-base CF and item-based CF. In our PF algorithm, the similar users adjusting parameter k_u and similar items adjusting parameter k_s have a great influence on the experimental results. In order to intuitively reveal experimental results, we let $k_u = k_s = k$, where k is the number of the nearest neighbor (users or items) used in the traditional collaborative filtering. The experimental results are shown in Figures 1 and 2.

As Figures 1 and 2 show, the item-base CF obviously has a lower value of MAE and RMSE than that of the user-based CF. However, the proposed method PF considers both user-based and item-based collaborative filtering, and utilizes the average of user's rating and item's rating. The results gained from the PF algorithm are better than the above two. When the neighborhood size is growing, the value of MAE is decreased. However, when the neighborhood size reaches to 80, we can see the value of MAE becomes bigger

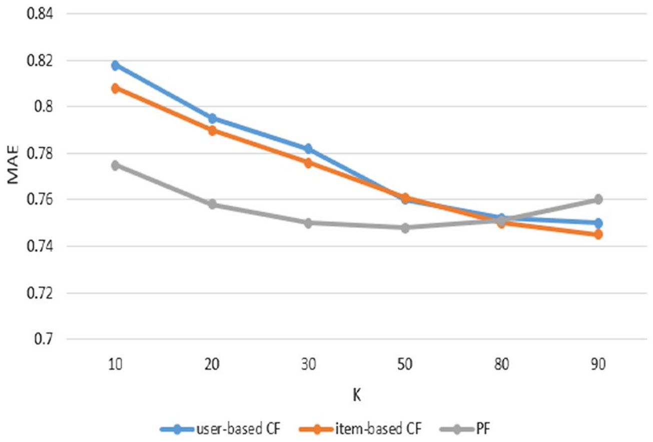

Figure 1 Comparison of PF algorithm and user-based CF, item-based CF with MAE.

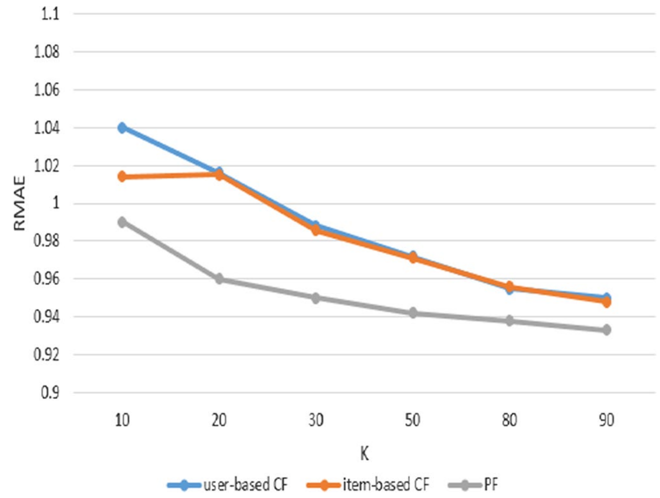

Figure 2 Comparison of PF algorithm and user-based CF, item-based CF with RMSE.

due to the over-fitting problem. The value of RMSE shows its stability when the neighborhood size is bigger than 50. And we can see that the PF algorithm curve is always below the user-based CF and the item-based CF, indicating that our proposed method achieves higher prediction accuracy. This is because that the average rating of users and items in our model are similarity.

We selected Radica basis function (RBF) as the kernel function for the SVM model and invoked the SVM toolbox in MATLAB 2008 directly. In order to verify the effectiveness of our deviation adjustment mechanism, we selected $k_u = k_s = k = 80$, for $k = 80$ is the best neighborhood size with regard to our PF algorithm. We also implemented the deviation adjustment by the BP neural network (BPNN). Detailed comparisons on MAE are shown in Table 3. From Table 3, we can see the deviation adjustment mechanism has indeed further lowered the value of MAE with both the traditional collaborative filtering and our PF. Compared with the used-based CF and item-based CF, the proposed method PF has higher predictive accuracy. Moreover, SVM model is better than BPNN model for the reason that SVR could reach global optimal.

Conclusion

In this paper, a personalized fitting recommendation approach has been proposed by combining the characteristic of the user's and item's similarity score set. We use it to predict the missing rating, and the results show great stability over a period of time for the reason of the average rating of users and items. Most traditional collaborative filtering methods only considered the rating data in the rating matrix. However, in this paper, we have further presented a deviation adjustment mechanism based on the SVR

Table 3 Comparing different deviation adjustment models with SVR by MAE

Method	BASIC	BPNN	SVR
User-based CF	0.760	0.756	0.750
Item-based CF	0.761	0.755	0.749
PF	0.745	0.743	0.740

by using the non-rating features. The experimental results have revealed that the non-rating attributes contributed to reduce the prediction errors.

In our future works, we will consider timeliness and optimize our algorithms to gain better personalized recommendation results.

Authors' contributions

WL proposed the idea of personalized fitting and drafted the manuscript. XL and MY designed and developed the algorithms of personalized fitting. MY conducted the experimental data collection and analysis. JJ designed the deviation adjustment. QJ supervised the work and critically revised the manuscript. All authors read and approved the final manuscript.

Author details

[1] School of Computer Engineering and Technology, Shanghai University, Shanghai, China. [2] Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai, China. [3] School of Computer, Beifang University of Nationalities, Yinchuan, Ningxia, China. [4] Faculty of Human Sciences, Waseda University, Tokorozawa, Japan.

Acknowledgements

This work is supported by Natural Science Foundation of Ningxia under Grant No. NZ12212.

Compliance with ethical standard

Competing interest

The authors declare that they have no competing interests.

Received: 2 October 2014 Accepted: 13 July 2015
Published online: 22 July 2015

References

1. Bank M, Franke J (2010) Social networks as data source for recommendation systems. E-commerce and web technologies. Lecture Notes in Business Information Processing, vol 61, pp 49–60
2. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intel 2009:19 **Article ID 421425**
3. Yu K, Schwaighofer A, Tresp V, Xiaowei Xu, Kriegel H-P (2004) Probabilistic memory-based collaborative filtering. IEEE Trans Knowl Data Eng 16(1):56–69
4. Jeong B, Lee J, Cho H (2010) Improving memory-based collaborative filtering via similarity updating and prediction modulation. Inf Sci 180(5):602–612
5. Pennock DM, Horvitz E, Lawrence S, Giles CL (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
6. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web (WWW '01). ACM, New York, pp 285–295
7. Konstan JA, Miller BN, Maltz D (1997) GroupLens: applying collaborative filtering to usenet news. Comm ACM 40(3):77–87
8. Brand M (2003) Fast online SVD revisions for lightweight recommender systems. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp 37–46
9. Zhou T, Ren J, Medo M, Zhang YC (2007) Bipartite network projection and personal recommendation. Phys Rev E 76(4):046115
10. Liu JG, Zhou T, Xuan ZG, Che HA, Wang BH, Zhang YC (2010) Degree correlation of bipartite network on personalized recommendation. Int J Mod Phys C 21(01):137–147
11. Fouss F, Pirotte A, Renders JM, Saerens M (2007) Random-walk computation of similarities between Nodes of a graph with application to collaboratvie recommendation. IEEE Trans Knowl Data Eng 19(3):355–369
12. Hallinan B, Striphas T (2014) Recommended for you: the Netflix Prize and the production of algorithmic culture. New Media Society, pp 1461444814538646
13. Li J, Wang X, Sun K, Ren J (2014) Recommendation algorithm with support vector regression based on user characteristics. In: Proceedings of the 9th international symposium on linear drives for industry applications, vol 3. Springer, Berlin, Heidelberg, pp 455–462
14. Pinheiro A, Cappelli C, Maciel C (2014) Increasing information auditability for social network users. In: Human interface and the management of information. Information and knowledge design and evaluation. Springer, pp 536–547
15. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749
16. Ahlgren P, Jarneving B, Rousseau R (2003) Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient. J Am Soc Inform Sci Technol 54(6):550–560
17. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings 14th Conference on Uncertainty in Artifical Intelligence (UAI), pp 43–52

18. Jeh G, Widom J (2002) SimRank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 538–543
19. Popescul R, Ungar LH (2001) Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001), Morgan Kaufmann, San Francisco, pp 437–444
20. Miyahara K, Pazzani MJ (2000) Collaborative filtering with the simple Bayesian classifier. In: Proceedings of the 6th Pacific International Conference on Artificial Intelligence. Berlin, pp 679–689
21. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 426–434
22. Hofmannn T, Puzicha J (1999) Latent class Models for Collaborative filtering, IJCA'99: Proceedings of the 16th International Joint Conference on Artificial Intelligence, pp 688–693
23. Wu YH, Chen YC, Chen ALP (2001) Enabling personalized recommendation on the Web based on user interests and behaviors, Research Issues in Data Engineering, 2001. In: Proceedings. Eleventh International Workshop on, pp 17–24
24. Hung L (2005) A personalized recommendation system based on product taxonomy for one-to-one marketing online. Expert Syst Appl 29(2):383–392
25. Zhang ZK, Zhou T, Zhang YC (2010) Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. Physica A 389(1):179–186
26. Shepitsen A, Gemmell J, Mobasher B, Burke R (2008) Personalized recommendation in social tagging systems using hierarchical clustering. In: Proceedings of the 2008 ACM conference on Recommender systems (RecSys '08). ACM, New York, pp 259–266
27. Song X, BL Tseng, CY Lin, MT Sun (2006) Personalized recommendation driven by information flow. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06). ACM, New York, pp 509–516
28. Baird L, Moore AW (1999) Gradient descent for general reinforcement learning. Adv Neural Inform Process Syst 11:968–974
29. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273
30. Vapnik V, Golowich SE, Smola AJ (1996) Support vector method for function approximation, regression estimation and signal processing. Advances in neural information processing systems. In: Proceedings of the 1996 Neural Information Processing Systems Conference NIPS 1996, pp 281–287
31. Xiao XF, Xu LH, Zhu Y (2012) Short-term traffic flow prediction based on SVM. J Guangxi Normal Univ Nat Sci Edn 30(4):13–17
32. Basak D, Pal S, Patranabis DC (2007) Support vector regression. Neural Inform Process Lett Rev 11(10):203–224
33. Wu CH, Ho JM, Lee DT (2004) Travel-time prediction with support vector regression. IEEE Trans Intel Transport Syst 5(4):276–281