

Exploit Rating Scale Model for Collaborative Filtering

Haijun Zhang*, Bo Zhang, Zhenping Li, Guicheng Shen

School of Information, Beijing Wuzi University, Beijing, China.

* Corresponding author. Tel.: +8613439061666; email: hjz_666@126.com

Manuscript submitted January 12, 2016; accepted April 12, 2016.

doi: 10.17706/jsw.11.6.528-537

Abstract: In this paper, the rating scale model is extended from one-dimension to multi-dimension, and then, a novel collaborative filtering algorithm is proposed. In this algorithm, user's interest is multi-dimensional, and item's quality that satisfies user's interest is multi-dimensional too. The rating of a user for an item is a weighted summation of all the latent ratings of the user for the item in all dimensions, and the weights at different interest dimensions are user-specific. The latent rating of user u for item i in one dimension is of a multinomial distribution which is determined by the user's interest value in this dimension, the item's quality value in this dimension, and the user's rating criteria. The parameters are estimated by minimizing the loss function using stochastic gradient descent method. Experimental results on benchmark datasets show that the algorithm has better performance than the compared algorithms.

Key words: Rating scale model, collaborative filtering, psychometrics.

1. Introduction

Collaborative filtering (CF) techniques can be categorized into two classes. One category is memory-based CF, and the other category is model-based CF [1], [2]. The recommendation of memory-based CF has good interpretability, and it is easy to convince users. Therefore, many companies use this CF technique, e.g. Amazon [3] and TiVo [4]. Memory-based CF includes user-based CF [5], [6] and item-based CF [7], [8]. User-based CF supposes that the test user will like the things that his or her similar users liked, and the estimated rating of the test user for the test item is adjusted by the ratings of his or her similar users. Item-based CF deems that the test user will like the other things similar to the things that he or she previously liked, and the estimated rating of the test user for the test item is calculated by the items this user rated. The key step of the memory-based CF is computing the similarity between users or items. The similarity is usually computed directly on user-item rating matrix. In this case, a row of the matrix is considered as the feature vector of a user, and a column of the matrix is considered as the feature vector of an item. The similarity between two vectors relies on the common elements, at which both of the two vectors have rating. If the data is very sparse, and the common rated elements between the two vectors are too few, it will result in the computed similarity incredible, and the low recommendation accuracy [9].

In order to alleviate the deviation of similarity resulted by data sparsity, Biyun Hu *et al.* [9], [10] first introduced psychometrics into CF. They use rating scale model of psychometrics to estimate user's latent interest from user-item rating matrix, and then the similarity between two users is estimated based on their latent interests. This algorithm alleviates the sparsity problem to some extent. But, it assumes that each user only has one interest and all the users have the same rating criteria. This does not conform to the reality, because user usually has many interests and has different rating criteria.

To solve this problem, in this paper, a novel collaborative filtering algorithm is proposed. This algorithm assumes that user's interest is multi-dimensional. The rating score is mainly determined by the satisfaction that the item brings to the user in all these dimensions, and it is also determined by the user's rating criteria. These assumptions are in line with the user's rating psychology.

The main contributions of this paper are as follows. First, the rating scale model is extended from one dimension to multiple dimensions, which can model user's multi-latent-interest and users' different rating criteria. Second, a method to learn the parameters in the model is proposed, and based on these parameters the ratings in test data set are computed directly. Our algorithm surpasses the algorithms proposed in [9], [10] because the latter need compute similarity between users and rely on memory-based CF to get the final predicted ratings. Third, experimental results show that our algorithm has higher prediction accuracy than the compared algorithms. Compared with our previous work [13], this paper extended the rating criteria to user-specific and BookCrossing dataset was added to evaluate the algorithms.

The rest of this paper is organized as follows. Section 2 describes the related work. In Section 3, the novel algorithm was proposed. The design of experiment and experimental results are presented in Section 4, followed by conclusions in Section 5.

2. Related Work

Biyun Hu *et al.* [9] first introduced the rating scale model into CF, which can be expressed by Eq. (1) and (2), where B_u denotes the quantitative measure of the interest of user u , and larger B_u means user u will give larger rating to items. Parameter D_i denotes the quantitative measure of shortcomings of item i , and smaller D_i means item i will get a larger rating from users. F_k is the rating criterion denoting the difficulty for users to give score k relative to give score $k-1$ to items, and P_{uik} is the probability for user u to rate item i with score k . Parameters B_u , D_i , and $F_2 \sim F_K$ need to be estimated using the training data. After these parameters are estimated, Biyun Hu *et al.* utilized B_u to compute similarity between users, and the similarity is computed by Eq. (3)-(5).

$$\log \frac{P_{uik}}{P_{ui(k-1)}} = B_u - D_i - F_k \quad k = 2, 3, \dots, K \quad (1)$$

$$F_1 = 0 \quad (2)$$

$$MaxInterest = \max_u (B_u) \quad (3)$$

$$MinInterest = \min_u (B_u) \quad (4)$$

$$s_{uv} = \frac{1 - |B_u - B_v|}{|MaxInterest - MinInterest|} \quad (5)$$

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in S_n(u)} (r_{vi} - \bar{r}_v) \cdot s_{uv}}{\sum_{v \in S_n(u)} |s_{uv}|} \quad (6)$$

In Eq. (5), s_{uv} is the similarity between user u and user v , B_u is the interest value of user u , and B_v is the interest value of user v . When all the similarities between test user u and other users have been computed, the predicted ratings can be computed by Eq. (6), where \hat{r}_{ui} is the estimated rating of user u for item i , \bar{r}_u is the average over all the ratings of user u in training set, r_{vi} is the true rating of user v for item i , s_{uv} is the similarity between user u and user v , and $S_n(u)$ is the set of n nearest neighbors of user u .

In another paper [10], Biyun Hu *et al.* proposed a novel method which has better performance than the previous one. Steps of the method are as follows. Firstly, the parameters in Eq. (1) are estimated using training set. Secondly, the expectation ratings are calculated by Eq. (9).

$$\psi_k = -\sum_{j=1}^k F_j \quad (7)$$

$$P_{uik} = \frac{e^{\psi_k + (k-1)(B_u - D_i)}}{\sum_{j=1}^K e^{\psi_j + (j-1)(B_u - D_i)}} \quad k = 1, 2, \dots, K \quad (8)$$

$$l_{ui} = \sum_{k=1}^K k \cdot P_{uik} \quad (9)$$

Equations (7)-(9) can be derived from Eq. (1)-(2). In Eq. (9), l_{ui} is the expectation rating of user u for item i . Since all the parameters have been obtained, all the expectation ratings of every user for every item can be computed by Eq. (7)-(9). In Biyun Hu's paper [10], when all the expectation ratings have been computed, the user-item expectation rating matrix can be obtained. A row in this matrix can be considered as the feature vector of a user. Then, based on these feature vectors, the n nearest neighbors of test user u can be obtained [10]. Finally, the estimated rating of test user u for item i can be calculated according to Eq. (6).

The limitations of these algorithms lie in the following aspects. First, the algorithms assume that a user has only one interest. But, actually, user usually has many interests. Second, after the parameters are estimated, these two algorithms need resort to memory-based CF to get the final predicted ratings, which causes its huge computation. Third, in Eq.(1), parameter F_k is user irrelevant, which means that all users have the same rating criteria. It is not reasonable because different user usually has different rating criteria.

3. Our Framework

In rating scale model expressed by Eq. (1), user's interest or item's quality is of one dimension. Actually, they should be multi-dimensional, because a user maybe has interest with value 2 in science fiction films, interest with value 8 in action movies, interest with value 1 in romantic films, interest with value 3 in movie scenes, and interest with value 5 in actors. As for a film, maybe 30% of its contents belong to science fiction and 20% belong to action genre. Its quality about scene may be quantified by value 2, and actors' performance is represented by value 3, and etc. The rating of a user for a movie is a comprehensive thought of satisfactions that the movie brings to the user in all these dimensions. Furthermore, the traditional rating scale model assumes that all the users have the same rating criteria, i.e. they share the same parameters F_k ($k=1, 2, \dots, K$). But, actually, different people may have different rating criteria. Based on this idea, the rating scale model is extended to multiple dimensions, which is expressed as Eq. (10)-(11).

$$\log \frac{P_{uikl}}{P_{ui(k-1)l}} = B_{ul} - D_{il} - F_{uk} \quad k = 2, 3, \dots, K; l = 1, 2, \dots, L \quad (10)$$

$$F_{u1} = 0 \quad (11)$$

where L is the number of interest dimensions, and it is a meta-parameter which should be determined beforehand. Parameter P_{uikl} is the probability that user u will give score k for item i in dimension l . B_{ul} is the interest value of user u in dimension l , and larger B_{ul} means user u has strong interest in dimension l , and user u will give a high rating for the item in this dimension, i.e. user u has a larger chance to give larger rating in dimension l . D_{il} is the quantitative measure of shortcomings of item i in dimension l , and larger D_{il}

means item i has little content related to dimension l and it will have a smaller probability to get a high rating in dimension l . F_{uk} is the rating criterion of user u which measures the difficulty to give score k relative to give score $k-1$ for an item. In Eq. (10), each rating has K -level scales. In MovieLens data set, the value of K is 5. The parameters B_{ul} , D_{il} , and F_{uk} need to be estimated using training set. For ease to expression, notation ψ_{uk} is introduced by Eq.(12), and then Eq. (14) can be deduced.

$$\psi_{uk} = -\sum_{j=1}^k F_{uj} \quad (12)$$

$$\psi_{u1} = 0 \quad (13)$$

$$P_{uikl} = \frac{e^{\psi_{uk} + (k-1)(B_{ul} - D_{il})}}{\sum_{j=1}^K e^{\psi_{uj} + (j-1)(B_{ul} - D_{il})}} \quad k = 1, 2, \dots, K ; \quad l = 1, 2, \dots, L \quad (14)$$

From Eq. (14), it can be seen that the rating of user u for item i in dimension l is of multinomial distribution with K categories, which is determined by B_{ul} , D_{il} , and ψ_{uk} . After P_{uikl} is obtained, the latent rating of user u for item i in dimension l can be computed by Eq. (15), and it is an expectation value. Then, the rating value of user u for item i can be computed by Eq. (16), where, $w_{ul} \geq 0$ is the weight in dimension l of user u , which will be estimated using training set.

$$\hat{r}_{uil} = \sum_{k=1}^K k \cdot P_{uikl} \quad (15)$$

$$\hat{r}_{ui} = \frac{\sum_{l=1}^L w_{ul} \cdot \hat{r}_{uil}}{\sum_{l=1}^L w_{ul}} \quad (16)$$

In order to learn the parameters in the model, a loss function was designed which is described as Eq. (17).

$$\begin{aligned} f &= \sum_{(u,i) \in Tr} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(\sum_{u,l} w_{ul}^2 + \sum_{u,l} B_{ul}^2 + \sum_{i,l} D_{il}^2 + \sum_{u,k} \psi_{uk}^2 \right) \\ &= \sum_{(u,i) \in Tr} \left(r_{ui} - \frac{\sum_{l=1}^L w_{ul} \cdot \hat{r}_{uil}}{\sum_{l=1}^L w_{ul}} \right)^2 + \lambda \left(\sum_{u,l} w_{ul}^2 + \sum_{u,l} B_{ul}^2 + \sum_{i,l} D_{il}^2 + \sum_{u,k} \psi_{uk}^2 \right) \\ &= \sum_{(u,i) \in Tr} \left(r_{ui} - \frac{\sum_{l=1}^L w_{ul} \cdot \left(\sum_{k=1}^K k \cdot \frac{e^{\psi_{uk} + (k-1)(B_{ul} - D_{il})}}{\sum_{j=1}^K e^{\psi_{uj} + (j-1)(B_{ul} - D_{il})}} \right)}{\sum_{l=1}^L w_{ul}} \right)^2 \\ &\quad + \lambda \left(\sum_{u,l} w_{ul}^2 + \sum_{u,l} B_{ul}^2 + \sum_{i,l} D_{il}^2 + \sum_{u,k} \psi_{uk}^2 \right) \end{aligned} \quad (17)$$

where, Tr denotes training set. The first term in the right side of the equation is the sum of squared error, and other terms are regularization factors applied to parameters to avoid over fitting. The gradient of each parameter is calculated by Eq. (20)-(23). For expression simplicity, notation e_{ui} and ψ_{kuil} are introduced by Eq. (18)-(19).

$$e_{ui} = r_{ui} - \hat{r}_{ui} \quad (18)$$

$$\psi_{kuil} = e^{\psi_{uk} + (k-1) \cdot (B_{ul} - D_{il})} \quad k = 1, 2, \dots, K ; l = 1, 2, \dots, L \quad (19)$$

$$\frac{\partial f}{\partial w_{ul}} = \sum_i 2e_{ui} \cdot \left(-\frac{\hat{r}_{uil} \cdot \sum_{l=1}^L w_{ul} - \sum_{l=1}^L (w_{ul} \cdot \hat{r}_{uil})}{\left(\sum_{l=1}^L w_{ul} \right)^2} \right) + 2\lambda w_{ul} \quad (20)$$

$$\frac{\partial f}{\partial B_{ul}} = \sum_i \left(2e_{ui} \cdot \frac{-1}{\sum_{l=1}^L w_{ul}} \cdot w_{ul} \cdot \sum_{k=1}^K k \cdot \frac{(k-1) \cdot \psi_{kuil} \cdot \left(\sum_{m=1}^K \psi_{muil} \right) - \psi_{kuil} \cdot \left(\sum_{m=1}^K (m-1) \cdot \psi_{muil} \right)}{\left(\sum_{m=1}^K \psi_{muil} \right)^2} \right) + 2\lambda B_{ul} \quad (21)$$

$$\frac{\partial f}{\partial D_{il}} = \sum_u \left(2e_{ui} \cdot \frac{-1}{\sum_{l=1}^L w_{ul}} \cdot w_{ul} \cdot \sum_{k=1}^K k \cdot \frac{(1-k) \cdot \psi_{kuil} \cdot \left(\sum_{m=1}^K \psi_{muil} \right) - \psi_{kuil} \cdot \left(\sum_{m=1}^K (1-m) \cdot \psi_{muil} \right)}{\left(\sum_{m=1}^K \psi_{muil} \right)^2} \right) + 2\lambda D_{il} \quad (22)$$

$$\frac{\partial f}{\partial \psi_{uk}} = \sum_{(u,i) \in Tr} \left(\frac{-2e_{ui}}{\sum_{l=1}^L w_{ul}} \cdot \sum_{l=1}^L \left(w_{ul} \cdot \frac{k \cdot \psi_{kuil} \cdot \left(\sum_{m=1}^K \psi_{muil} \right) - \psi_{kuil} \cdot \left(\sum_{m=1}^K m \cdot \psi_{muil} \right)}{\left(\sum_{m=1}^K \psi_{muil} \right)^2} \right) \right) + 2\lambda \psi_{uk} \quad (23)$$

In Eq. (18), r_{ui} is the true rating value of user u for item i . As it is shown in Eq. (20)-(23), all the partial derivatives of the loss function are elementary functions. Therefore, all the partial derivatives of the loss function are continuous, and hence the loss function is differentiable. Thus stochastic gradient descent method [11, 12] can be used to search the satisfactory solution. The update rules are described by Eq.(24)-(27).

$$w_{ul} \leftarrow \max \left(0, w_{ul} + \gamma_1 \left(e_{ui} \cdot \left(\frac{\hat{r}_{uil} \cdot \sum_{l=1}^L w_{ul} - \sum_{l=1}^L (w_{ul} \cdot \hat{r}_{uil})}{\left(\sum_{l=1}^L w_{ul} \right)^2} \right) - \lambda w_{ul} \right) \right) \quad (24)$$

$$B_{ul} \leftarrow B_{ul} + \gamma_2 \left(\frac{e_{ui}}{\sum_{l=1}^L w_{ul}} \cdot w_{ul} \cdot \sum_{k=1}^K k \cdot \frac{(k-1) \cdot \psi_{kuil} \cdot \left(\sum_{m=1}^K \psi_{muil} \right) - \psi_{kuil} \cdot \left(\sum_{m=1}^K (m-1) \cdot \psi_{muil} \right)}{\left(\sum_{m=1}^K \psi_{muil} \right)^2} - \lambda B_{ul} \right) \quad (25)$$

$$D_{il} \leftarrow D_{il} + \gamma_2 \left(\frac{e_{ui}}{\sum_{l=1}^L w_{ul}} \cdot w_{ul} \cdot \sum_{k=1}^K k \cdot \frac{(1-k) \cdot \psi_{kuil} \cdot \left(\sum_{m=1}^K \psi_{muil} \right) - \psi_{kuil} \cdot \left(\sum_{m=1}^K (1-m) \cdot \psi_{muil} \right)}{\left(\sum_{m=1}^K \psi_{muil} \right)^2} - \lambda D_{il} \right) \quad (26)$$

$$\psi_{uk} \leftarrow \psi_{uk} + \gamma_3 \left(\frac{e_{ui}}{\sum_{l=1}^L w_{ul}} \cdot \sum_{l=1}^L w_{ul} \cdot \frac{k \cdot \psi_{kuil} \cdot \left(\sum_{m=1}^K \psi_{muil} \right) - \left(\sum_{m=1}^K m \cdot \psi_{muil} \right) \cdot \psi_{kuil}}{\left(\sum_{m=1}^K \psi_{muil} \right)^2} - \lambda \psi_{uk} \right) \quad (27)$$

where the initial values of learning rate parameters $\gamma_1, \gamma_2, \gamma_3$ and the initial value of regularization parameter λ will be tuned by held out data. For expression simplicity, let's introduce some vector notations: $B_u = \{B_{u1}, B_{u2}, \dots, B_{uL}\}^T$, $w_u = \{w_{u1}, w_{u2}, \dots, w_{uL}\}^T$, $\psi_u = \{\psi_{u1}, \psi_{u2}, \dots, \psi_{uK}\}^T$, and $D_i = \{D_{i1}, D_{i2}, \dots, D_{iL}\}^T$. Finally, the workflow of our algorithm CFMRSM (Collaborative Filtering based on Multidimensional Rating Scale Model) is shown in follows. After the parameters are obtained, the ratings in test set can be estimated by Eq. (16).

Algorithm CfmrsM. Collaborative Filtering based on Multidimensional Rating Scale Model

Input: Training set Tr

Output: Vectors B_u, D_i, w_u , and ψ_u

1. Initialize parameters $B_u, D_i, w_u, \psi_u, L, \gamma_1, \gamma_2, \gamma_3$, and λ .
 2. For each iteration
 3. Randomly permute the data in Tr .
 4. For each r_{ui} in Tr
 5. Step 1. Compute e_{ui} by Eq.(18);
 6. Step 2. Compute ψ_{kuil} ($k=1, 2, \dots, K; l=1, 2, \dots, L$) by Eq.(19);
 7. Step 3. Update the parameters according to Eq.(24)-(27).
 8. End for
 9. Decrease $\gamma_1, \gamma_2, \gamma_3$; they are multiplied by 0.9.
 10. End for
-

4. Experiments

4.1. Dataset and Evaluation Metric

MovieLens¹ dataset and BookCrossing² dataset are used to evaluate the algorithm CFMRSM. MovieLens data set contains 100000 rating scores (1-5 scales) rated by 943 users on 1682 items, where each user has more than 20 ratings. BookCrossing data set contains 278,858 users and 1,149,780 ratings (scales 0-10) about 271,379 books, and the ratings are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0. We only extract the explicit ratings to get a dataset which contains 77,805 users and 433,671 ratings about 185,835 books. The user-item ratings of each data set are randomly split into 5 equal parts with no overlap. They will be used for 5-fold cross validation. Four parts of them are used as training set and the other one is used as test set. In addition, we use a held out data set to tune the initialization parameters, and when the values of these parameters are determined, they will not be changed in the later cross validation tests.

RMSE (root mean squared error) and MAE (mean absolute error) are used as evaluation metrics.

¹<http://www.grouplens.org/>

²<http://www.informatik.uni-freiburg.de/~chiegler/BX/>

Formally,

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in Te} (r_{ui} - \hat{r}_{ui})^2}{|Te|}} \quad (28)$$

$$MAE = \frac{\sum_{(u,i) \in Te} |r_{ui} - \hat{r}_{ui}|}{|Te|} \quad (29)$$

where $|Te|$ is the number of tested ratings in test set Te , \hat{r}_{ui} is the estimated rating of user u for item i , and r_{ui} is the true rating value of user u for item i . The lower the RMSE or MAE, the better the performance is.

4.2. Algorithm Evaluation

4.2.1. Performance comparison

The compared algorithms include user-based CF, item-based CF, RSVD[11], and collaborative filtering based on one dimensional rating scale model (named as CFRSM) which is proposed in [10]. RSVD is a very strong baseline method, which achieved great success in Netflix competition. Except for user-based and item-based CF methods, each algorithm was run 5 times on every test set independently, and we report the average results over 25 times. The comparison results are shown in table 1. For algorithm CFMRSM, the initial values of B_{ui} and D_{ij} are sampled from uniform distribution on $[0, 1]$, the initial value of w_{ui} is set to 1, and ψ_{uk} is set to 0. For MovieLens data set, the values of γ_1, γ_2 and γ_3 are set with 0.05, 1.0 and 0.05, respectively, and the iteration times on training set is 100. For BookCrossing data set, γ_1 is set with 0.05, γ_2 is set with 0.02, γ_3 is set with 0.5, and the iteration times on training set is 5. For both data set MovieLens and BookCrossing, the parameter λ is set with 0.001, and L is set with 20. The number of latent factors in algorithm RSVD is set with 100, the algorithm is iterated 100 times on the training set, and the parameters are tuned similar to paper [11].

From Table 1 and Table 2, it can be seen that, on both data sets, algorithm CFMRSM has lower RMSE and MAE than algorithm CFRSM, User-based CF, and Item-based CF. It means that when user's interests are extended from one dimension to multiple dimension and rating criteria is set related to specific user, the algorithm can model the data better. Meanwhile, it can be observed from Table 1 that algorithm CFMRSM shows a slight lower precision than RSVD on MovieLens data set, but it shows a significant higher precision than RSVD on BookCrossing data set. It is because that the ratings in BookCrossing data set have 10 level scales which is larger than 5 level scales in MovieLens data. Algorithm CFMRSM have more parameters to model users' rating criteria on BookCrossing data set, so it achieved significantly better performance on BookCrossing data set. It means that modeling users' rating criteria by algorithm CFMRSM is effective.

Table 1. Comparison on RMSE with Compared Algorithms

DataSet	CFMRSM	RSVD	CFRSM	User-based CF	Item-based CF
MovieLens	0.9192	0.9141	0.9363	0.9633	0.9415
BookCrossing	1.6351	1.7830	1.7498	1.8351	1.8584

Table 2. Comparison on MAE with Compared Algorithms

DataSet	CFMRSM	RSVD	CFRSM	User-based CF	Item-based CF
MovieLens	0.7235	0.7184	0.7250	0.7570	0.7410
BookCrossing	1.2462	1.3587	1.3275	1.3895	1.4013

4.2.2. Sensitivity analysis of the parameters

From Section 3, we can see that, the algorithm CFMRSM extended the rating scale model at two aspects: first, user's interest is extended from one dimension to multiple dimension; second, rating criteria are extended to user-specific. In this section, some experiments are designed to study the necessity of these extensions.

If the rating criteria are not extended to user-specific, the model can be expressed by Eq. (30)-(31), which is named as WPRC (Without Personal Rating Criteria) proposed in our previous work [13]. In this model, if user's interest is not extended to multi-dimension, i.e. the value of parameter L is set to 1, it will degenerated to traditional rating scale model expressed by Eq.(1)-(2).

$$\log \frac{P_{uikl}}{P_{ui(k-1)l}} = B_{ul} - D_{il} - F_k \quad k = 2, 3, \dots, K; l = 1, 2, \dots, L \quad (30)$$

$$F_1 = 0 \quad (31)$$

Book Crossing data set is used to analyze the sensitivity of the user-specific rating criteria and multi-dimensional interests. The results are shown in Fig. 1 and Fig. 2. As shown in Fig. 1 and Fig. 2, with the increase of the parameter L , the algorithm WPRC and CFMRSM have improved performance. Therefore, it can be concluded that, the extension of user's interest from one dimension to multiple dimension brings improved performance. At the same time, algorithm CFMRSM always has better performance than WPRC, which means that it is meaningful to extend rating criteria (i.e. parameter F_k) to user-specific (i.e. F_{uk}).

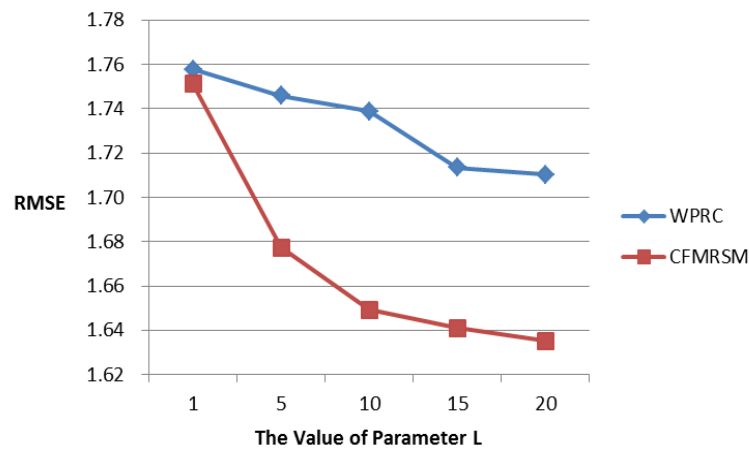


Fig. 1. The sensitivity analysis of the parameters on RMSE.

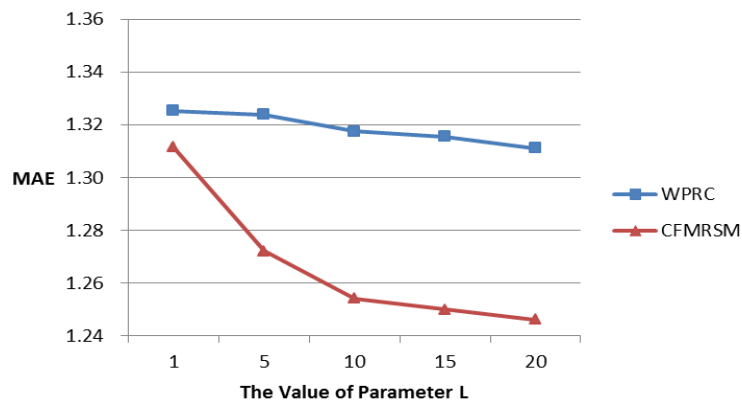


Fig. 2. The sensitivity analysis of the parameters on MAE.

5. Conclusions

In this paper, a novel collaborative filtering algorithm based on rating scale model is proposed, which extended the rating scale model from one dimension to multiple dimension. Meanwhile, a method based on stochastic gradient descent is proposed to learn the parameters in the model. Experimental results show that, it outperforms compared algorithms including the algorithm based on one dimensional rating scale model, user-based collaborative filtering, and item-based collaborative filtering. It also surpasses the state-of-art algorithm—RSVD on BookCrossing data set. In the future, we will study how to fuse this algorithm with other algorithms to improve the performance further.

Acknowledgment

This work is supported by NSFC (No. 71540028), the Fund of Beijing Key Laboratory [No. BZ0211], the Fund of Beijing Social Science [No.14JGC103], the Statistics Research Project of National Bureau [No. 2013LY055], and the High-level Research Projects' Cultivation Fund of Beijing Wuzi University (No. GJB20141002).

References

- [1] Gediminas, A., & Alexander, T. (2005). Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, 634–749.
- [2] Su, X. Y., & Taghi, M. K. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 1–19.
- [3] Greg, L., Brent, S., & Jeremy, Y. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7, 76–80.
- [4] Ali, K., & Van, S. W. (2004). TiVo: Making show recommendations using a distributed collaborative filtering architecture. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (pp. 394–401).
- [5] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Commun*, 40(3), 77-87.
- [6] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Commun. ACM*, 40(3), 56-58.
- [7] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the WWW Conference*.
- [8] Deshpande, M., & Karypis, G. (2004). Item-based top-w recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1), 143-177.
- [9] Hu, B. Y., Li, Z. J., & Wang, J. User's latent interest-based collaborative filtering. 619–622.
- [10] Hu, B. Y., Li, Z. J., & Chao, W. H., *et al.* User preference representation based on psychometric models. *Proceedings of the 22nd Australasian Database Conference (ADC 2011)*.
- [11] Koren, Y., Bell, R., Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 2009, 42(8), 30-37.
- [12] Deepak, A., & Bee-Chung, C. (2009). Regression-based latent factor models. 19-27.
- [13] Zhang, H., Zhang, X., Li, Z., *et al.* Collaborative filtering using multidimensional psychometrics model. 691-697.



Haijun Zhang was born in Henan province, China, in 1975. He received the B.Sc. degree, and the M.Sc. degree in management science from the China University of Mining and Technology, in 1997, 2004 respectively. His major interests are data mining and

recommendation system. He is now an associate professor in School of Information at Beijing Wuzi University.



Bo Zhang was born in Liaoning province, China, in 1959. He received the B.Sc. degree in automation engineering from the Liaoning University of Technology, in 1982. He got his master degree in logistics engineering from Beijing Wuzi University in 2009. He is currently an associate professor in School of Information at Beijing Wuzi University. His major interests are electronic commerce and computer network.



Zhenping Li was born in Shandong province, China, in 1966. She received her bachelor's degree in operations research from Shandong University of China in 1989. In 1994, she received her master's degree in operations research from Shandong University. In 2004, she received her doctoral degree in operations research from Chinese Academy of Science. She majored in combinatorial optimization and was interested in the complex network and intelligent algorithm. She is now a professor in School of Information at Beijing Wuzi

University.



Guicheng Shen was born in Jiangsu province, China, in 1966. He got his doctor degree in computer science and technologies from Beijing University of Science and Technology in June, 2009. He got his master degree and bachelor degree in mathematics from Beijing Normal University in 1993 and 1990 respectively. He is now a professor in School of Information Beijing Wuzi University. His major research directions are data mining, business intelligence and information security.