Recommended Items Rating Prediction based on RBF Neural Network Optimized by PSO Algorithm

Chengfang Tan, Caiyin Wang, Yulin Li and Xixi Qi

***Abstract*—**In order to mitigate the data sparsity and cold-start problems of recommendation item ratings and more accurately predict the ratings of recommended items, according to the characteristics of the non-linearity and randomness in recommendation content changes, recommended items ratings prediction model based on RBF neural network optimized by PSO algorithm is proposed and the parameters of RBF neural network are globally optimized by using the proposed PSO algorithm. Experimental results show that, compared with the state-of-the-art models including the traditional user-based collaborating filtering method, the traditional item-based collaborating filtering method and RBF neural network model, the proposed RBF neural network optimized by PSO algorithm can more accurately predict the recommendation item ratings, which has higher prediction accuracy and much lower prediction error measured by Root Mean Square Error (RMSE), Precision@N and Recall@N.

***Keywords*—**PSO algorithm, RBF neural network, recommended item rating prediction, data sparsity.

I. INTRODUCTION

The rapid development of Internet triggers the problem of information overload. As an important means to solve the problem of information overload, the recommendation system is widely used in the fields of electronic commerce, mainly filtering out the unrelated items to alleviate the burden of users. At present, the recommendation system can be roughly categorized into the content-based recommendation system and the collaborative filtering recommendation system. However, one of the biggest problems existing in the collaborative filtering recommendation is that, as most users are not willing to provide item rating information or new items do not have enough rating data, the clustering effect from similar users is poor [1,2]. That is to say, under the inaccurate clustering class of similar users, the cross recommendation among users will seriously affect the recommendation quality and even produce wrong recommendation, ultimately, data sparsity and cold start problems are produced.

Aiming at the above mentioned problems of collaborative filtering recommendation systems, many scholars have put forward their views from different angles, and have achieved good results [3]. For example, reference 4 fused the cosine similarity and Pearson correlation with Jaccard similarity coefficients method for the accurate measurement of the similarity between users. In reference 5, the user's social relation is introduced into the recommendation system to realize the better recommendation. Reference 6 combined the trust relationship between users with the collaborative filtering recommendation, utilized the explicit trust relationship and the transition of trust relation to calculate the trust degree between users, and selected the nearest neighbor based on the strength of trust relationship. Reference 7 integrated the matrix factorization model with the trust relationship between users, exploited the influence and the transitivity of trust relationship to map users and items into the same feature space, then the preference prediction of users was performed in the feature space. According to reference 8, the personal information of users could reflect the user's interest, and put forward adopting the user's personal information to complement the user's preference. Reference 9 put forward the concept of the same scoring matrix from users, based on the similarity between the users and the same scoring matrix, the not evaluated items were performed the preference prediction

Motivated by the above observations, we propose a new recommended items rating prediction method based on RBF (Radial Basis Function) neural network optimized by PSO algorithm (short for PSO-RBF). PSO algorithm not only has a strong global search capability, but also is easy to be achieved. The parameters of RBF neural network optimized by PSO algorithm are the global optimal parameters, which can overcome the problem of low reliability of RBF neural network learning and guarantee that the recommended items ratings are more accurately predicted.

This work was supported by Key Projects of Excellent Young Talent Support Program in Colleges and Universities of Anhui Province (gxyqZD2016344) ,University Students Innovative Undertaking Training Program of Anhui Province (201710379021, 201710379023), Teaching Research Project of Suzhou University (szxy2016jy32), the Key Projects of Anhui Province Colleges and Universities Natural Science Foundation of China (KJ2016A768) and Soft Science Research Project of Anhui Science and Technology Plan in 2016 (1607a0202071).

Chengfang Tan is with School of Information Engineering, Suzhou University, Suzhou 234000, Anhui, China (corresponding author; e-mail: 874036730@qq.com).

Caiyin Wang is with School of Information Engineering, Suzhou University, Suzhou 234000, Anhui, China.

Yulin Li is with School of Information Engineering, Suzhou University, Suzhou 234000, Anhui, China.

Xixi Qi is with School of Information Engineering, Suzhou University, Suzhou 234000, Anhui, China.

II. The Proposed Recommended Items Rating Prediction based on RBF Neural Network Optimized by PSO Algorithm

In the following sections, we first introduce particle swarm optimization algorithm and RBF neural network described, respectively. Then based on RBF neural network optimize by PSO algorithm, a PSO-RBF based recommended items rating prediction model is presented in detail. Lastly, we reports experimental results.

1. Particle swarm optimization algorithm

Particle Swarm Optimization algorithm [10] is an evolutionary algorithm based on swarm intelligence proposed by Kennedy and Eberhart in 1995. In PSO algorithm, the feasible solution of each optimization problem is regarded as a particle in the search space. Assume that there are N particles in motion in D dimensional space, $v_i = (v_{i,1}, v_{i,2} \cdots v_{i,D})$ denotes the velocity vector of particle i, $X_i = (x_{i,1}, x_{i,2} \cdots x_{i,D})$ represents the space position vector of particle i. Particle i constantly adjusts its speed and space position according to v_i and X_i until it reaches the global optimal solution. P_{best} denotes the optimal solution of particle itself and $P_{best} = (p_{i,1}, p_{i,2} \cdots p_{i,D})$. The optimal solution for the population at present is expressed by g_{best} and $g_{best} = (g_{i,1}, g_{i,2} \cdots g_{i,D})$. $v_i^{(t)} = (v_{i,1}{}^t, v_{i,2}{}^t \cdots v_{i,D}{}^t)$ is the velocity of particle i in time t, then the speed of i in the $t+1-th$ iteration is expressed as follows:

$$v_i^{(t+1)} = wv_i^t + c_1 rand(\,)(P_{bi} - X_i^{(t)}) + c_2 rand(\,)(g_{bi} - X_i^{(t)}) \quad (1)$$

Where w is the inertia factor and its value is greater than 0, c_1 is cognitive learning factor and c_2 is social learning factor, c_1 and c_2 are both positive constants, $rand(\,)$ is a function that generates the random number ranging from 0 to 1. The velocity value of the particle is in the range of $[v_{\min}, v_{\max}]$. If the velocity of a particle is less than $v_{\min}$ or greater than $v_{\max}$, then the velocity value of the particle is set to the corresponding threshold value. In the $t+1-th$ iteration, the position of particle i is denoted as shown in Equation (2),

$$X_i^{(t+1)} = X_i^{\,t} + v_i^{(t+1)} \quad (2)$$

During actually utilizing PSO algorithm, if the number of iterations reaches the maximum value or the particle finally finds the best optimal location in accordance with the minimum fitness threshold, then iteration is terminated [11].

2. RBF neural network

In 1998, Broomhead et al. proposed a radial basis function based neural network (short for RBF) [12]. Similar with other neural networks, RBF neural network has a good local approximation, fast convergence and so on. The structure of RBF neural network is shown in Fig. 1 as follows:

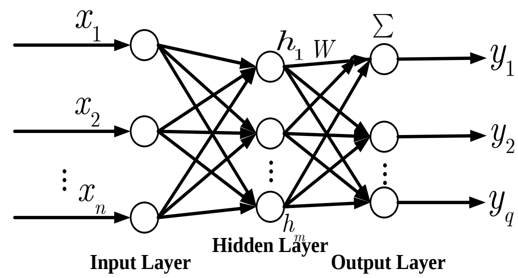

Fig. 1. Structure of RBF neural network

From Fig. 1, it can be observed that RBF neural network includes input layer, hidden layer and output layer. It is a feed forward neural network with the structure of three layers. Input layer can achieve the function of simply transferring the input data into the hidden layer, hidden layer neurons are able to make the input data produce non-linear mapping. Subsequently, neurons in the output layer perform linear weighted operations towards the nonlinear data output from the hidden layer, and ultimately results are output in a linear style. As shown in Fig. 1, $X = [x_1, x_2, \cdots, x_n]^T$ is an n-dimensional input vector, $H = [h_1, h_2, \cdots, h_m]^T$ is a radial basis functions used in the hidden layer, which denotes the output of hidden layer unit and is usually represented by using the Gauss function. The equation is presented as follows,

$$h_i(x) = \exp\left(-\frac{\| X - C_i \|}{2b_i^2}\right) \quad i = 1, 2, \cdots, m \quad (3)$$

In Equation (3), C_i represents the center of the first $i-th$ basis function, b_i denotes the width of the $i-th$ basis function, and b_i is a number greater than 0. $\|\bullet\|$ is the Euclidean paradigm, m denotes the number of nodes in the hidden layer. $W = [w_1, w_2, \cdots, w_m]^T$ is the weighted vector of RBF neural network. The output of RBF neural network is a linear combination of the output of the hidden layer nodes and is denoted as,

$$y_k(x) = \sum_{k=1}^{q} W_{ik} h_i(x) \quad (4)$$

In Equation (4), $k = 1, 2, \cdots, q$ and q is the number of nodes in the output layer, W_{ik} represents the connected weights between the $i-th$ node from the hidden layer and the node from the output layer.

From the description on RBF neural network, it can be visible that, in the RBF neural network $k-th$, Gauss basis function center vector C_i, Gauss basis function width vector b_i, the number of radial basis function center m and the connected weights W_{ik} between the output layer and the hidden layer need to be determined. After determining the above mentioned C_i, b_i and m, the least squares method can be used to directly compute the connected weights W_{ik} between the output layer and the hidden layer. C_i, b_i and W_{ik}

determine the performance of the whole RBF neural network together.

3. The proposed recommended items rating prediction model

From the description of RBF neural network, it can be concluded that the prediction performance of RBF neural network mainly depends on Gauss basis function center vector C_i, Gauss basis function width vector b_i, and connected weights W_{ik} between the output layer and the hidden layer. However, the traditional RBF neural network adopts the local information based on the parameter space to set the relative parameters, and it causes that the values of parameter C_i, b_i and w_{ik} are the local optimal solutions, not the global optimal solutions. Aiming at the above mentioned disadvantages from RBF neural network, during the process of predicting the changes of recommendation items rating, we make use of PSO algorithm to perform the global optimization towards three parameters C_i, b_i and w_{ik} in RBF neural network. The proposed recommended items rating prediction algorithm based on RBF neural network optimized by PSO algorithm (short for PSO-RBF) is introduced in detail as follows:

Step 1: Initialize the particle swarm and neural network. The pre-processed recommendation samples are fed into RBF neural network for training, and three above mentioned parameters C_i, b_i ,and W_{ik} are obtained. Parameters C_i, b_i and w_{ik} are respectively regarded as the initial position of the particle, the velocity of the particle and the position vector of the particle to form one by one particle.

Step 2. Compute and evaluate the fitness value of each particle. Suppose v_i and $\hat{v}_i$ respectively denote the true value and the predictive value of the recommended item rating at a certain time. Assume that σ is the number of training examples, Fitness function $adpt$ is defined to measure the merits of the selection parameters and its equation is denoted as follows,

$$adpt = -\frac{1}{\sigma}\sum_{i=1}^{\infty}\left|\frac{v_i - \hat{v}_i}{v_i}\right| \tag{5}$$

From Equation (5), it can be represented that the negative mean relative error between the true value v_i and the predictive value $\hat{v}_i$ of the recommended item rating is regard as a measure of the fitness value of each particle.

Step 3. Update the search position of each particle according to the fitness value of each individual, and calculate the individual extreme value $pbest$ and the population global extreme value $gbest$ of each particle. The detailed operations are as follows, for each particle, we compare the fitness value of each particle with the best position $pbest$ that this particle has ever experienced, if the current fitness value of this particle is better, and then the current fitness value of the particle is regarded as the current best position $pbest$. For each particle, we compare the fitness value of each particle with the population global extreme $gbest$ that all particles have ever experienced, if it is better, and then $gbest$ is set to the current position of the best particle.

Step 4. Utilizing the characteristics of PSO algorithm, the three parameters C_i, b_i and w_{ik} in RBF neural network are continuously performed globally optimized. If the iteration reaches the maximum number or the mean square error reaches the initial setting value, then the particle search is terminated and the optimal particle position is output; otherwise returns step 3, the iterative optimization is executed repeatedly, until the optimal values of the parameters are obtained in RBF neural network.

Step 5. Input the measured data to predict the change of the recommended items rating. The decoded value of the best position that the swarm has experienced is regarded as the structural parameter of RBF neural network, and the decoded value is taken into RBF neural network. We utilize another group of the recommended items rating to test the trained recommended items rating prediction model and the prediction efficiency of the recommended items is obtained.

III. Experimental Results and Verification

In this section, we conduct comparative experiments to evaluate the effectiveness and efficiency of the proposed PSO-RBF based recommended items rating prediction model. Specifically, we aim to answer the following two questions:

(1) Can the proposed PSO-RBF based recommendation method improve the recommendation performance by accurately predicting the rating of recommended items?

(2) Is the proposed PSO-RBF based recommended method able to mitigate the cold-start problem for recommendation by accurately predicting the rating of items?

We begin by introducing two real-world datasets, the experimental settings and the evaluation metrics we choose, then we compare the PSO-RBF based recommendation method with the three state-of-the-art baselines to answer the first question and we explore the capability of the proposed PSO-RBF method in handling the cold-start problem to answer the second question. All the algorithms were implemented using Java and all experiments were performed on a server running Windows Server 2008 with four 3.00GHz CPU cores and 12GB memory. In this paper, we exploit the same ordinary data structures for all the algorithms and do not consider any parallel computation.

1. Experimental datasets

We collect two real-world datasets for the comparative experiments, among which, one is from the social media website Epinions (http://www.Epinions.com) that provides the online services for users, users can perform many information sharing operations on Epinions, for example, reviewing and rating the items, etc. From the originally collected datasets, we filter out users who rated few items and also items that received less than 10 ratings. The final adopted dataset from Epinions is shown in Table 1.

Table 1. Statistical information on Epinios Dataset

Dataset	Number of users	Number of items	Number of ratings
Epinions	40,163	139,738	664,824

Another dataset is from Movielens, which includes about 100 million movie rating information that 6,040 users rated the 3,952 movies with scores from 1 to 5. The statistics of the MovieLens dataset is shown in Table 2.

Table 2. Statistical information on MovieLens Dataset

Dataset	Number of users	Number of movies	Number of ratings
MovieLens	6,040	3,952	About 100M

2. Evaluation metrics and experimental setting

(1) Evaluation Metrics

Firstly, we adopt the Root Mean Squared Error (RMSE) to measure the prediction error. RMSE is defined as follows [13]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i,j}(\hat{r}_{ij} - r_{ij})^2} \quad (6)$$

Where r_{ij} denotes the rating of recommendation item j by user i, $\hat{r}_{ij}$ denotes the corresponding rating predicted by the model, and N denotes the total number of the test set. The smaller the value of RMSE, the more precise a recommendation is.

Secondly, ranking a recommendation list is often more important than the rating prediction, so we also evaluate the algorithms in terms of ranking. To measure the accuracy of ranking a recommendation list, we adopt two standard metrics Precision@N (denoted by P@N) and Recall@N (denoted by R@N). Specifically, we think that a item rating is successfully predicted if it is in the recommendation list. P@N defines the ratio of successfully predicted item topics to the N recommendations, and R@N defines the ratio of successfully predicted item topics to the number of item topics to be predicted. Generally, P@N and R@N are computed by the following equations:

$$P@N = \frac{N_{it}}{N} \quad (7)$$

$$R@N = \frac{N_{it}}{N_{it} + \bar{N}_{it}} \quad (8)$$

Where N is the number of recommended items, N_{it} is the number of items contained in both the ground truth and the top-N results, $\bar{N}_{it}$ denotes the number of items contained in the ground truth but not in the top-N results. Note that P@N and R@N are the averages of precision values and recall values, respectively. We will demonstrate how the performance of recommendation models varies with different values of N in the following subsection E.

(2) Experimental Setting

To investigate the capability of the proposed PSO-RBF based recommendation method, for both two datasets Epinions and Movielens, we random select 40%, 60% and 80% as the training set respectively and the corresponding remaining 60%, 40% and 20% as the testing set. The random selection process is carried out 10 times independently, and the average RMSE are obtained. Note that parameters of the proposed PSO-RBF based recommendation method are determined via cross validation.

3. Baseline methods

We compare the proposed PSO-RBF method with the three baseline recommendation methods. Specifically, the three baseline methods are presented as follows,

(1) UserCF: UserCF refers to the user-based collaborative filtering method and is a memory-based recommendation method. UserCF mainly adopts the neighborhood information of users in the user-item rating matrix for supporting the whole recommendation process. In this paper, we use the cosine similarity to calculate user-user similarity.

(2) ItemCF: ItemCF refers to the item-based collaborative filtering method and is also a memory-based recommendation method. ItemCF mainly adopts the neighborhood information of items in the user-item rating matrix for supporting the whole recommendation process. In this paper, we use the Pearson correlation to calculate item-item similarity.

(3) RBF neural network based method: RBF neural network is one of the neural networks. About the detailed information of RBF neural network, please refer to the above mentioned introduction on RBF neural network.

4. Performance comparisons in terms of RMSE

To answer the first question referred to in the beginning of this section, we compare the proposed PSO-RBF based method with the above three baseline methods on two real-world datasets. The comparative results are shown in Table 3 on RMSE.

Table 3. Performance comparison on Epinions and Movielens in terms of RMSE

Dataset	Training size	UserCF	RBF based	PSO-RBF based
Epinions	40%	1.3240	1.3145	**1.2522**
	60%	1.3015	1.2803	**1.2365**
	80%	1.2861	1.2543	**1.2190**
MovieLens	40%	1.2876	1.2654	**1.2358**
	60%	1.2653	1.2521	**1.2180**
	80%	1.2430	1.2382	**1.1853**

From the comparative results in Table 3, we can conclude the following observations: In general, under three different training sizes, ItemCF method always outperforms UserCF method, and compared to other three baseline methods, the proposed PSO-RBF based method always obtains better performance than other three baseline methods and is more robust to the data sparsity problem. These results indicate that RBF neural network optimized by PSO algorithm can further improve the recommendation item rating prediction performance. With the increasing size of the training set, we have enough experimental data to train every model and every model becomes more stable, hence, the performances of the PSO-RBF based method and other three baseline methods are all improved on RMSE to some extent.

Via aforementioned analysis, we can draw an answer to the first question that RBF neural network optimized by PSO algorithm not only can significantly improve the recommendation performance but also can mitigate the data sparsity problem in the recommendation systems. In summary, the proposed PSO-RBF recommendation method is superior to the simple collaborative filtering method, because training from RBF neural network optimized by PSO algorithm filtered out the rating values with greater evaluation bias.

5. Performance comparisons in terms of P@N and R@N

In order to further verify the effectiveness of utilizing PSO algorithm to optimize RBF neural network for evaluating recommendation lists, through comparative experiments, P@N and R@N are utilized to sort the predicted items according to the ratings and the values of N are set to 10, 15 and 20, respectively. The comparative experimental results are presented in Fig. 2, Fig. 3, Fig. 4 and Fig. 5 as follows.

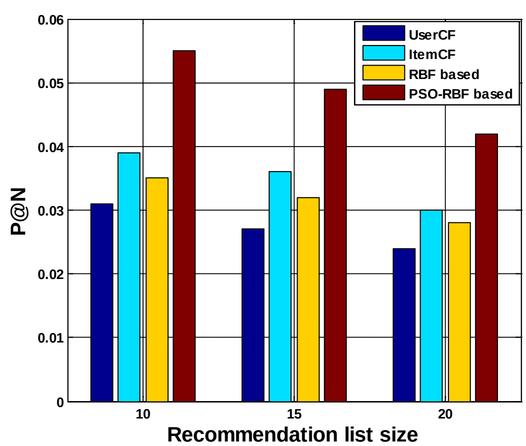

Fig. 2. P@N performance comparison on Epinions dataset (N=10, 15 and 20 respectively)

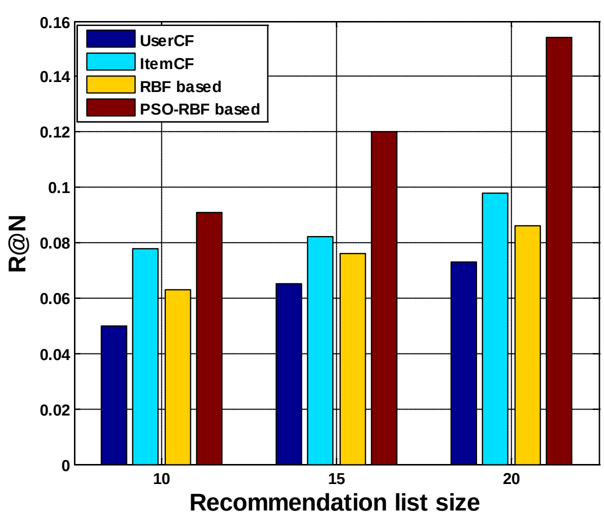

Fig. 3. R@N performance comparison on Epinions dataset (N=10, 15 and 20 respectively)

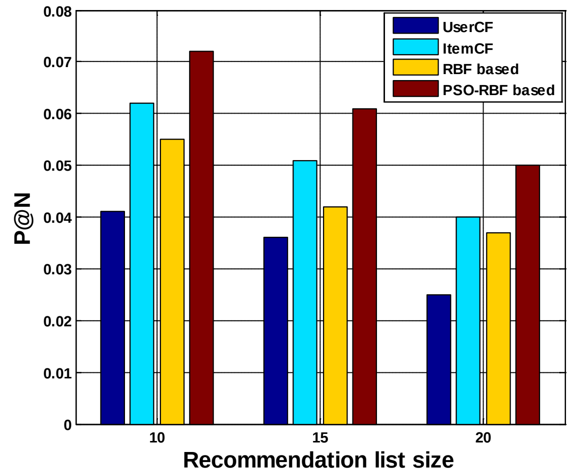

Fig. 4. P@N performance comparison on Movielens dataset (N=10, 15 and 20 respectively)

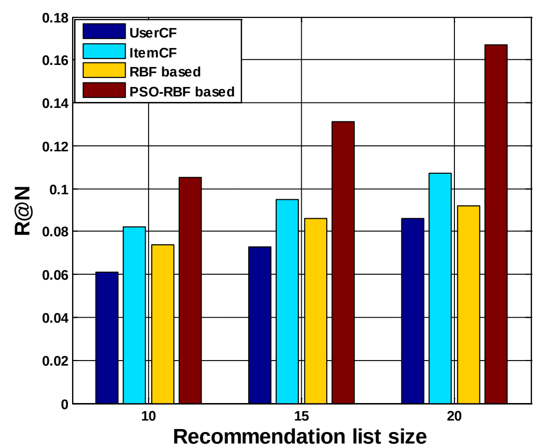

Fig. 5. R@N Performance comparison on Movielens dataset (N=10, 15 and 20 respectively)

We start with discussing the results showed in Fig. 2, Fig. 3, Fig. 4 and Fig. 5. The changes of all recommendation models with varying recommendation list size on Epinions and Movielens are presented. Obviously, for all models, larger recommendation list size decreases the precision but increases the recall. That is to say, the larger the recommendation list size, the lower the precision is and the higher the recall is. This is because larger recommendation list size increases the probability that a user's item ranking is hit by one of the recommended items. The maximum recommendation list size is set to 20, because larger size may affects the effective interaction between a user and the item. In addition, when the value of N is fixed, our proposed PSO-RBF based recommendation approach obviously outperforms UserCF, ItemCF and RBF based method in terms of recall and precision. We carefully compare the values of P@N and R@N, for example, on Epinions dataset, the value of P@10 from the PSO-RBF based method is the largest compared with UserCF method, ItemCF method and RBF based method in Fig. 2. On Movielens dataset, the value of R@10 from the PSO-RBF based method is also the largest compared with UserCF method, ItemCF method and RBF based method in Fig. 5. And other comparisons about the values of P@N and R@N are also similar. This is because predicting the missing ratings of items makes the PSO-RBF based recommendation method be able to accurately calculate the similarity between items, so as to improve the accuracy of the recommendation item ranking prediction. Hence, we can conclude that the idea of compensation values for items missing rating can improve the accuracy of the recommendation lists, and mitigates the data sparsity and cold-start problems in the traditional recommendation systems to some extent.

6. Ability of coping with cold-start items

To answer the second question mentioned in the beginning of Section 4, we investigate the capability of the proposed PSO-RBF method in handling cold-start problems. In detail, similar with the above distribution of the training set and the test set, we randomly select 8% items from the training set and remove their item ratings from the training set to the test set. In this way, these 8% of items do not have any item ratings and we consider these 8% of items as cold-start items. The results on those training sets with cold-start items are listed in Table 4 for RMSE.

Table 4. Performance comparison on Epinions and Movielens with 8% cold-start items in terms of RMSE

Dataset	Training size	UserCF	ItemCF	RBF-based	PSO-RBF based
Epinions	40%	1.5621 (-0.2381)	1.4850 (-0.1990)	1.5342 (-0.2197)	1.3601 (-0.1079)
	60%	1.5310 (-0.2295)	1.4507 (-0.1986)	1.5129 (-0.2326)	1.3327 (-0.0962)
	80%	1.4681 (-0.1820)	1.4258 (-0.1971)	1.4532 (-0.1989)	1.3038 (-0.0848)
MovieLens	40%	1.4670 (-0.1794)	1.4460 (-0.2008)	1.4603 (-0.1949)	1.3520 (-0.1162)
	60%	1.4192 (-0.1539)	1.4076 (-0.1726)	1.4136 (-0.1615)	1.3201 (-0.1021)
	80%	1.3830 (-0.1400)	1.3651 (-0.1416	1.3705 (-0.1323)	1.2836 (-0.0983)

Note that numbers inside parentheses in Table 4 denote the performance reductions compared to the performance without cold-start users in Table 3. We analyze Table 4 and compare the performance with Table 3 without cold-start items, it can be concluded that the performance of all methods degenerates when we introduce cold-start items. For example, when training size is 80%, RMSE for PSO-RBF based method increases up to 1.3038 from 1.2190 in Table 3 on Epinions, that is to say, RMSE decreases by 1.2190-1.3038= -0.0848. RMSE for PSO-RBF based method increases up to 1.2836 from 1.1853 in Table 3 on Movielens, that is to say, RMSE decreases by 1.1853-1.2836= -0.0983. We carefully observe Table 3 and Table 4, the performance degeneration of the proposed PSO-RBF based recommendation method is much smaller compared to the other three baseline methods. These results support that the proposed PSO-RBF based recommendation method can mitigate cold-start problems for the recommendation. In summary, the introduction of cold-start items could degrade the recommendation performance and the proposed framework is relatively more robust to cold-start items by predicting the missing ratings of items.

IV. Conclusion

RBF neural network optimized by PSO algorithm is a regularized network and can approach any continuous function, which has a strong predictive power. In this paper, RBF neural network optimized by PSO algorithm is designed to accurately predict the recommended items ratings, which is different from calculating the user similarity with the same items' ratings by users in the traditional user based recommendation method. Compared with UserCF, ItemCF, and RBF-based method, the predictive result of the proposed PSO-RBF based method is more accurate. The comparative results also show that the PSO-RBF based model has better stability and is relatively more robust to cold-start items by predicting the missing ratings of items than the existing recommended item rating prediction. Our next research work will incorporate the similarity of items, the similarity of users and the social relationship among users into the PSO-RBF based recommendation model, and construct the correlations among the similarity of items, the similarity of users and the social relationship among users which further improve the accuracy of the PSO-RBF based recommendation model.

References

[1] M. Deshpande, G. Karypis, "Item-based Top-n Recommendation Algorithms", *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143-177, 2004

[2] Y. Koren, R. Bell, C. Volinsky, "Matrix Factorization Techniques for Recommender Systems", *Computer*, vol. 42, no. 8, pp. 30-37, 2009

[3] J. Kim, D. Lee, K. Chung, "Item Recommendation based on Context-aware Model for Personalized U-healthcare Service", *Multimedia Tools and Applications*, vol.71, no. 2, pp. 855-872, 2014

[4] K. Choi, Y. Suh, "A New Similarity Function for Selecting Neighbors for Each Target Item in Collaborative Filtering", *Knowledge-Based System*, vol. 37, no.1, pp.146-153, 2013

[5] C. Cheng, H. Yang, I. King, M. R. Lyu, "Fused Matrix Factorization with Geographical and Social Influence in Location-based Social Networks", *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 17-23, 2012

[6] Y. Du, L. Huang, M. Ho, "A Collaborative Filtering Recommendation Method Fusion Trust Computing", *Pattern Recognition and Artificial Intelligent*, vol. 5, no. 1, pp. 417-425, 2014

[7] M. Duan, "Collaborative Filtering Recommendation Algorithm based on Trust Propagation", *International Journal of Security and Its Application*, vol. 9, no. 7, pp. 99-108,2015

[8] C. Laia, D. Liub, C. Linb, "Novel Personal and Group-based Trust Models in Collaborative Filtering for Document Recommendation", *Information Science*, pp.31-49, 2013

[9] S. Rendle, L. S. Thieme, "Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation", *Proceedings of the third ACM International Conference on Web Search and Data Mining*, pp.81-90, 2010

[10] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Networks*, pp.1942-1948, 1995

[11] S. Solomon, P. Thulasiraman, R. Thulasiram, "Collaborative Multi-swarm PSO for Task Matching Using Graphics Processing Units", *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp.1563-1570, 2011

[12] H. G. Han, Q. L. Chen, J. F. Qiao, “An Efficient Self-organizing RBF Neural Network for Water Quality Prediction”, *Neural Networks*, vol. 24, no. 7, pp.717-725, 2011

[13] S. H. Wang, J. L. Tang, H. Liu, “Toward Dual Roles of Users in Recommender Systems”, *Proceedings of the 24th ACM International Conference on Information and Knowledge Manage*., pp.1651-1660, 2015

Chengfang Tan was born on Feb. 16, 1981. She received the Master degree in computer science and technology from Nanjing Normal University of China. Currently, she is a researcher (assistant professor) at Suzhou University, China. Her major research interests include information processing and data mining. She has published many papers in related fields.

Caiyin Wang was born on Sept. 16, 1977. He received the Master degree in computer science and technology from Hefei Polytechnic University, China. . His major research interests include artificial intelligence. He has published many papers in related fields.

Yulin Li is an undergraduate student from the School of Computer Science and Technology, Suzhou University. Her research interests include machine learning and network congestion control.

Xixi Qi is an undergraduate student from the School of Computer Science and Technology, Suzhou University. Her research interests include computer networks and security, machine learning, and intelligent information processing.