DYNAMIC RECOMMENDATION BASED ON USERS' LONG-TERM AND SHORT-TERM PREFERENCES AND SOCIAL IMPACT

[1]LI FEI DA, [1]YIN SI ZHE, [1]HAN SONG HE, [1]LIU YU JING, [2]SUN HAO

[1]School of Computer Science and Technology, China University of Mining and Technology, Xuzhou Jiangsu 221116, China

[2]International Business Division, China Telecommunications, Beijing, 10010, China

E-mail: life1730@foxmail.com

ABSTRACT

Aiming at the irrationality of existing social recommendation algorithms, such as not fully mining the preference correlation between users and not fully considering the dynamic change of user preferences in time, this paper proposes a dynamic recommendation algorithm based on user long-term and short-term preferences and social influence. From the dynamic change of user preferences and the social relationship of users, the improved gating cycle unit is used to model the user rating information and extract the long-term and short-term preference features; According to the user correlation matrix, the social impact of learning target users is expressed. Experiments on two real datasets show that the proposed algorithm is better than the comparison algorithm.

Keywords: *Social Recommendation, Preference Learning, Collaborative Filtering, GRU, Recommendation Algorithm*

1. INTRODUCTION

Introducing social relationships into the recommendation system can effectively alleviate problems such as data sparsity and cold start. Social relationships may be made to the user's information supplement, enrich user information, and measure users in many ways, thereby improving the effectiveness of the recommendation system. Compared with traditional recommendation systems that usually only focus on users-items, social recommendation with social network information not only needs to consider the simple binary relationship between users and items, but also consider the trusting and preference relationship between users and users [1]. Although these algorithms have good recommendation effects, they have some common limitations. Most algorithms assume that social networks are homogeneous, that is, connected users in social networks have the same preferences. In fact, the users just have the same preference in some ways, but vary between different users on different projects. In addition, most algorithms ignore the dynamic changes of the target user's time preferences.

In response to the above problems, this paper proposes a dynamic recommendation based on users' long-term and short-term preferences and social impact(DR-UPSI). Through the user correlation matrix module to learn the social influence between users and friends, in-depth study in different aspects of the user's different friends on the user's preferences, and considering the user's preferences will change over time, this paper improves the traditional gated recurrent unit (GRU), considers the key information of the time interval between consecutive user rating data, and predicts the user's long- and short-term preferences. The combination of the two achieves a more accurate recommendation, and experiments on two real data sets have proved that the method proposed in this paper is superior to the comparison algorithm.

2. RELATED WORK

In recent years, the rapid development of the social media has created a good development prospect for social recommendation and further promoted the development of this field. Yang et al. aims to integrate the sparse rating data of users and the social relationships between users. Using matrix factorization technology, users are mapped to low-dimensional latent feature spaces based on their social relationships, which more accurately reflects the mutual influence of users on the formation of self-views, and learns better user preference models to obtain high-quality recommendations [2]. Peng et al. proposed a recommendation algorithm(SPMF) that sets different recommendation trust weights according to the preferences of different user relationships in order to distinguish the different preferences of users with social relationships [3]. Guo et al. adds social trust information on the basis of improved singular value decomposition(SVD++) [5]. The two ways of influencing trust will have an impact on users, ensuring that in the case of few or no ratings, it is still possible to learn the implicit vector of the user from the trust information, and at the same time alleviate the problem of data sparseness and cold boot to a certain extent [4]. Yan et al. proposed a new relationship network fitting algorithm to control the spread and shrinkage of social information relationships, and establish a separate relationship network for each user and project. Then the matrix factorization and social regularization methods are combined, and the neighborhood model of individual relations is used to generate recommendations [6]. The social information-assisted recommendation algorithms proposed in the above paper have alleviated the two major problems of the recommendation system to a certain extent, but there are also some problems: the algorithms mostly ignore the hidden vector of the item; user preferences are not static; lack of follow-up ability to acquire deep and complex nonlinear features in social relationships.

In response to the above-mentioned problems, the [7] is based on collaborative filtering, taking into account the user's interest and the periodicity of the project to synthesize, and then obtain the recommendation list. In recent years, the successful application of deep learning in the recommendation field has also brought huge changes to the recommendation system. The recommendation system based on deep learning can effectively capture the non-linear interaction of user items, extract deep features from complex input information, overcome the limitations of traditional collaborative filtering, and achieve high-quality recommendation performance. Xue et al. proposed a deep learning architecture to learn a common low-dimensional space for the representation of users and items [8]. Fan et al. inputs the social network relationship into the graph embedding model (node2vec), and then obtains the low-dimensional representation of the social user, and integrates it into the probability matrix decomposition for scoring prediction [9]. Fan et al. proposed a deep social collaborative filtering framework(DSCF), which makes full use of social information and extracts useful information from the social relationships of multi-hop neighbors for recommendation [10]. Pan et al. proposed a Sparse Stacked Noise Reduction Autoencoder(SSDAE) to solve the problem of sparse and unbalanced data in social networks [11]. Chen et al. integrates the user's trust relationship, the user's distrust relationship, and the similarity between the user's trust relationship and the project into the probabilistic matrix factorization model, and proposes an enhanced matrix factorization technology that integrates three social factors. Recommendation model(EnSocialMF), comprehensively considers recommendations to users [12].

How to model user preferences based on existing user behavior information is a very important issue in recommendation algorithms. He et al. builds a user preference matrix based on existing user rating data and item category data. The users with similar preferences are clustered through the user preference matrix, and then the user-based collaborative filtering algorithm is used to make recommendations. In fact, explicit behaviors such as user ratings and comments are also very sparse. Compared with the display behavior, the user's implicit behavior is richer [13]. Qiu et al. uses two typical user behaviors, watching and liking, as an auxiliary tool to enhance recommendation. At the same time, combined with buying behavior, the trinity predicts user preferences and makes recommendations better [14]. Furthermore, they classify items into different types according to the received actions. Then on the basis of analyzing the co-occurrence of different types of behaviors, quantifying the correlation between them, it can examine the differences in user preferences between different types of projects. However, the above work does not take into account the dynamic changes of users' preferences [15].

The social relationship between users has huge potential value in the recommendation system, so social recommendation is getting more and more attention from researchers. The general principle behind most current social recommendation algorithms is that the preferences of users are similar to or influenced by users who have social relationships with them. However, treating all social relationships equally may result in a decrease in recommendation performance. In this regard, Le et al. organically combines the user's social network and the user-item internal connection to make social recommendations [16]. Wang et al. based on the matrix factorization method, weights the social trust relationship between users differently, and adds the similar relationship between users to the social trust

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

relationship as a supplement to enhance the calculation of user neighborhoods [17]. Pal et al. proposed a new path based on trust reasoning method, which uses the implicit influence information in existing social networks to calculate the trustworthiness of users to reconstruct the social relationships of users [18]. Li et al. learns the preferences between users on the basis of noticing neural networks, and uses network embedding technology as pre-training to integrate the extracted factors into the model for recommendation [19].

The sequence information of user behavior reflects the change of user preferences. As a powerful tool for processing sequence data, Recurrent Neural Network (RNN) is more suitable for recommendation systems. The general process of RNN is: the network memorizes the previous information and applies it to the calculation of the current output; at the same time, the nodes between the hidden layers are connected; the input of the hidden layer includes not only the input of the input layer, but also the previous one. Chai et al. based on the cyclic neural network sets up the time transition matrix and the space transition matrix to model the user's time and space preference information respectively, and comprehensively consider the sequence of continuous check-in points of interest and other information to predict the next behavior of the user [20]. But when the sequence data is too long, RNN will have the problem of disappearing gradient, which will seriously affect the results. GRU is an improved network of RNN. It solves the problem of gradient disappearance by adding gating units, and has a simpler structure and better performance. However, RNN and GRU do not consider the time interval between sequences, and this time interval does reflect the key information of user preferences. In order to solve the problem, this paper improves the GRU to process the sequence data of user behaviors, and uses the time interval information between behaviors.

3. DR-UPSI

This section will introduce the proposed model in detail. The overall model is shown in Figure 1, from bottom to top, from data input to prediction result output. Among them, the prediction of long-term and short-term user preference and the learning of user social influence of the core key part are shown in 3.2 and 3.3.

3.1 Model and problem description

Suppose that the user set is denoted by U= {U_1, U_2, ..., $U_{|M|}$}, and the item set is denoted by V = {V_1, V_2, ..., $V_{|N|}$}, let I=□ ^{M*N} denote the user- item interaction matrix, symbol M is the number of users and symbol N is the number of items. i_{u1v1}=1 indicates that there is interaction between user U_1 and project V_1, otherwise i_{u1v1}=0. Meanwhile, the social matrix between users is expressed as S=□ ^{M*M}, S_{u1u2}=1 indicates that there is a social relationship between users U_1 and U_2, otherwise it is 0. Given user set U, item set V, social matrix S and user item interaction matrix I, the goal of the model is to predict users' preferences for unrated items, and then select the top k list of predicted rated items.

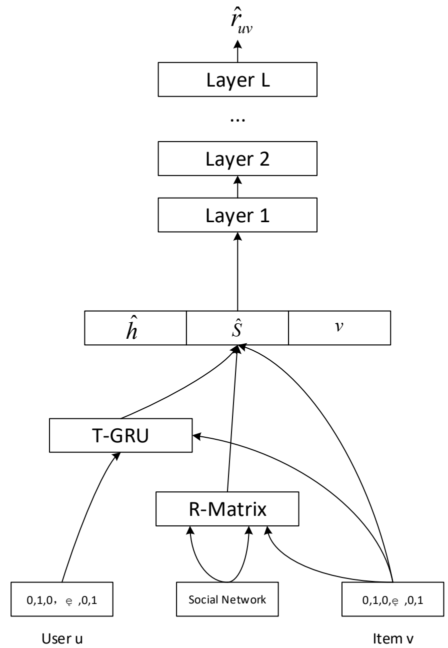

Figure 1: model structure

3.2 Users' long-term and short-term preference learning

GRU is a powerful and effective tool in processing sequence data. The calculation rules of traditional GRU at each moment are shown in Formula 1-4.

$$z_s = \sigma_z(x_s W_{xz} + h_{s-1} W_{hz}), \quad (1)$$

$$r_s = \sigma_r(x_s W_{xr} + h_{s-1} W_{hr}), \quad (2)$$

$$\tilde{h}_s = \tanh(x_s W_{xh} + (r_s h_{s-1}) W_{hh}), \quad (3)$$

$$h_s = z_s \odot h_{s-1} + (1 - z_s) \odot \tilde{h}_s, \quad (4)$$

Where, x_s is the input at the current moment, z_s is the update gate, r_s is the reset gate, σ is the activation function sigmoid, tanh() is the hyperbolic tangent activation function, $\tilde{h}_s$ is the candidate hidden state,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

h_s is the hidden state of the output, W_{xz}、W_{hz}、W_{xr}、W_{hr}、W_{xh} and W_{hh} are the weight matrix. GRU During sequence data modeling, r_s determines what to remove by multiplying the previous moment state h_{s-1}. If r_s is close to 0, then GRU chooses to forget the past data and only retain the current input r_s, while z_s selects the data of candidate hidden state $\tilde{h}_s$ and the data in h_{s-1} at the previous moment. If z_s is close to 1, then $1-z_s$ is close to 0, then the GRU will retain most of h_{s-1} and ignore most of $\tilde{h}_s$.

In this paper, we use the user's score on the item in continuous time as the user's whole behavior sequence data. In life, the user's rating data in the recent time can best represent the user's current preference, while the rating data with a long time span is difficult to represent the user's current preference.

However, the time interval information between continuous user behaviors is not considered in the traditional GRU. Moreover, the time interval information between user behaviors is an important information to reflect the long-term and short-term preferences of users. In order to use this time interval information on GRU, this paper adds a time gate to the traditional GRU to deal with the time interval information between two user behaviors. Thus, a gated cycle unit (Time-GRU) with time interval information is proposed, as shown in Figure 2.

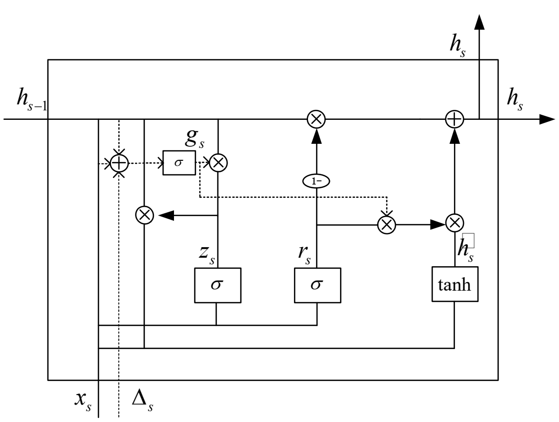

Figure 2: Time-GRU

The dotted line part is the work done on the basis of traditional GRU. The time gate g_s is determined by x_s, h_{s-1} and time interval Δ_s, as shown in formula 5.

$$g_s = \sigma_g(x_s W_{xg} + h_{s-1} W_{hg} + \sigma_{\Delta_s}(\Delta_s W_{sg})), \tag{5}$$

Where, Δ_s is the time interval between the current user behavior and the last behavior,σ is the activation function, and W_{xg}, W_{hg} and W_{sg} are the weight matrix. Then the new $\tilde{h}_s$ is as shown in formula 6.

$$\tilde{h}_s = \mathrm{Tanh}(x_s W_{xh} + (g_s \odot r_s \odot h_{s-1}) W_{hh}), \tag{6}$$

Where W_{xh} and W_{hh} are weight matrices. If the time interval Δ_s is small, then g_s is also small. then the information of h_{s-1} has little impact on the current x_s, and vice versa Similarly, the new h_s is shown in formula 7.

$$h_s = g_s \odot z_s \odot h_{s-1} + (1 - g_s \odot z_s) \odot \tilde{h}_s, \tag{7}$$

Then the output representation of time s is as shown in formula 8.

$$\hat{c}_s = \tanh(h_s), \tag{8}$$

It can be seen from the above formula that the size of s represents the time interval between user behaviors in continuous time. A small value of s indicates that the user's short-term preference has not changed significantly, and vice versa. $\hat{c}_s$ Indicates the preference change of the user s in the current state. Therefore, through Time-GRU, users' long-term and short-term preferences can be dynamically obtained.

3.3 User social impact

In user social networks, each user is connected to each other, and the general potential preferences between users affect each other. In other words, users' preferences will be influenced by their friends. This is also in line with real life. When people choose projects, they are more inclined to ask their friends for advice. Therefore, the feature vector of friends is regarded as the attention vector to guide users, that is, the social influence of users. When users interact with different projects, this social impact should be dynamic. Because on different projects, users will ask different friends for advice, and different friends have different effects on users. On the basis of reference [19], given the user vector and friend vector, this paper first learns the user correlation matrix R-Matrix, as shown in formula 9.

$$R_u = \mathrm{ReLU}(\tilde{u}_u F_u^T W_r), \tag{9}$$

Where, $\tilde{u}_u$ is the vector of user u, Fu is the matrix of f_u of user u's friend vector, and W_r is the weight. The user correlation matrix R serves as the preference similarity between users and their friends, that is, the social impact affecting user preferences. Then, when the user interacts with different items, the attention vector between the user and his friends is shown in formula 10-13.

Journal of Theoretical and Applied Information Technology
31st October 2021. Vol.99. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$H^{f_u} = \mathrm{Re}LU(F_u W_f + (\tilde{u}_u W_u)R^T + v_v W_v), \tag{10}$$

$$a^{f_u} = soft\max(H^{f_u} W_h{}^T{}_{f_u}), \tag{11}$$

$$H^{u_u} = \mathrm{Re}LU(\tilde{u}_u W_u + (F_u W_f)R^T + v_v W_v), \tag{12}$$

$$a^{u_u} = soft\max(H^{u_u} W_h{}^T{}_{u_u}), \tag{13}$$

Where, v_v is the item vector, W_f, W_u, W_v, $W_h^T{}_{f_u}$ are the weight matrix, $H^{f_u}$is the attention score learned by converting the vectors of users' friends and projects into the same potential space. a^{f_u} is the attention vector of friends when users interact with different items. Similarly, H^{u_u} is the preference score and a^{u_u} is the user preference vector. Therefore, the social influence vector affecting user preferences is shown in formula 14.

$$\hat{S}_u = \sum a^{u_u} \tilde{u}_u + \sum a^{f_u} f_u, \tag{14}$$

The $\hat{S}_u$ obtained from the above calculation is the influence characteristics of friends on user behavior when users interact with different projects according to the historical interaction between friends and projects. In other words, when users interact with different projects, users are also affected differently based on social interaction, which is adaptive.

3.4 Fusion prediction

The fusion layer uses addition to fuse the feature vector representing the user's long-term and short-term preferences obtained from 3.2 and the user's social impact vector obtained from 3.3 to obtain the user's item based representation and social based representation as a complete representation affecting the user's preferences. In the prediction task, the fused representation is input to the full connection layer to guess the user's score, as shown in formula 15.

$$h_l = \sigma_l(W_l(\sigma_{l-1}(W_{l-1}...\sigma_1(W_1 h_0 + b_1) + h_{l-1})) + b_l), \tag{15}$$

Where h_0 is the feature representation after fusion, l is the number of layers, i is the activation function, W_l is the weight and b_l is the offset. The final prediction user u's score on item v is shown in formula 16,

$$\hat{y}_{uv} = \sigma(W_l^T h_l). \tag{16}$$

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Dataset

In order to evaluate the effectiveness of DR-UPSI, this paper selects two data sets: CiaoDVD data set and Douban film data set. The data set contains user scores, user social relations, scoring time and other data required by this model. The user scores in both data sets are the scores of [1,5]. Detailed statistical information is shown in Table 1.

In the process of model verification, this paper randomly selects 80% as the training set, 10% as the verification set and the remaining 10% as the test set in the two data sets. The data set is preprocessed, and the user set is U, the item set is V, the user-item interaction set is I, the time set is T, and the social set is S.

Table 1: Ciaodvd dataset and Douban dataset

Data set	CiaoDVD	Douban
users	17,615	2,648
items	16,121	44,586
ratings	72,664	58,487
social relation	40,133	91,768

4.2 Metrics

In this paper, the common evaluation indexes of recommendation system are selected.

Hit rate (HR): HR @ K measures whether a test item appears in the top K items of the predicted recommendation list. If so, it is a hit. Normalized impairment cumulative gain(NDCG): NDCG@K Measure whether the test items appear in the top K items, focusing on order. Therefore, the higher the value of the two indicators, the better the effect of the model. The calculation formula is shown in formula 17 and formula 18.

$$HR@K = \frac{\sum Hits@K}{|TN|}, \tag{17}$$

$$NDCG@K = Z_k \sum \frac{2^{r_i} - 1}{\log_2(i+1)}, \tag{18}$$

Where |TN| is the number of test sets, and the numerator is the cumulative number of test sets in the first K item list of each user. r_i represents the correlation at i. if the item at i is in the test set, r_i is 1, otherwise it is 0. Z_k is the coefficient, which represents the reciprocal of the latter summation formula in the best case. The value range of NDCG is 0-1.

4.3 Comparison algorithm

In order to verify the effectiveness of the proposed algorithm, the following algorithm is selected as the comparison algorithm.

ITCF: Based on collaborative filtering, user interest and project periodicity are simply and linearly fused [7].

SDMF: A depth matrix decomposition model proposed by Xue et al. Learns a common low dimensional space for the representation of users and projects [8].

CNSR: A neural network architecture proposed by Le et al. Organically combines the internal relationship between user social network and user project for social recommendation [16].

ScAN: A social recommendation method proposed by Li et al. Learns the preferences between users on the basis of paying attention to neural networks, uses network embedding technology as pre-training, and integrates the extracted factors into the model for recommendation [19].

4.4 Experiment analysis

This paper makes a comparative experiment on CiaoDVD data set and Douban data set. The comparison algorithms include ITCF algorithm, DMF algorithm, CNSR algorithm and scan algorithm. Evaluation index selection hit rate HR@K Cumulative increase of normalized impairment NDCG@K , K values are 1, 5, 10, 15, 20.

The experimental results are shown in Figure 3-Figure 6.

Figure 3 and Figure 4 show the HR of ITCF, DMF, CNSR , ScAN and DR-UPSR in different recommendation list lengths on CiaoDvD dataset and Douban dataset.

Figure 5 and Figure 6 show NDCG of ITCF, DMF , CNSR, ScAN and DR-UPSI in different recommended list lengths on CiaoDvD dataset and Douban dataset.

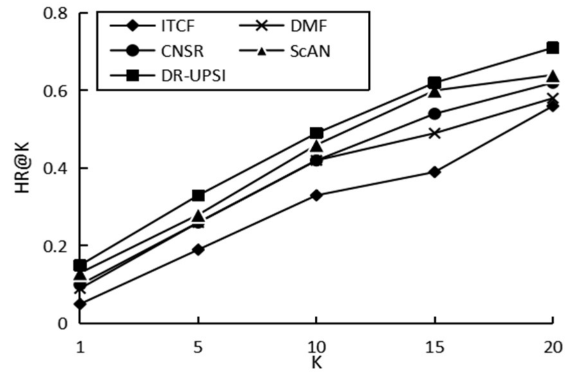

Figure 3: Different K values of different algorithms on ciaodvd dataset HR@K

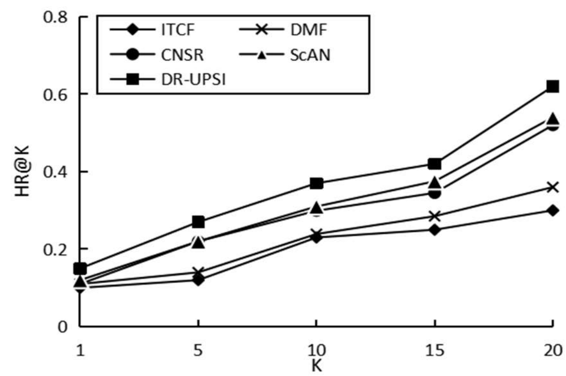

Figure 4: Different K values of different algorithms on Douban dataset HR@K

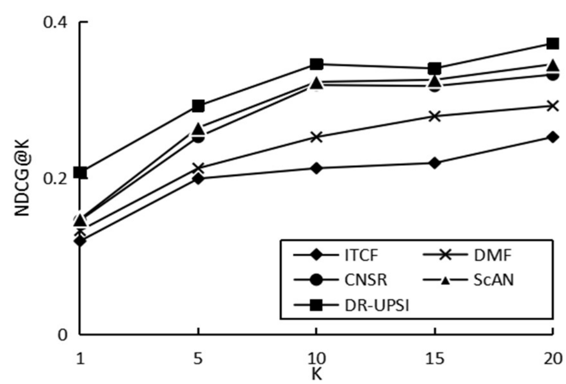

Figure 5: Different K values of different algorithms on ciaodvd dataset NDCG@K

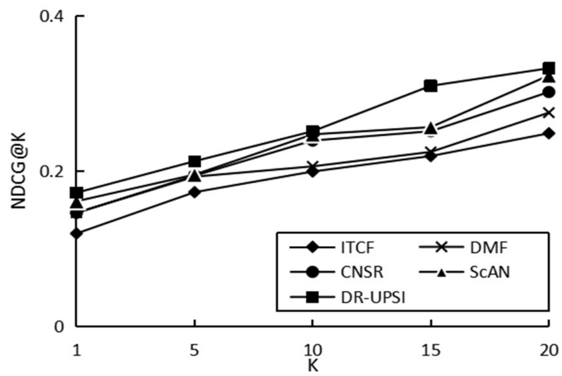

Figure 6: Different K values of different algorithms on Douban dataset NDCG@K

From the above results, the following conclusions can be drawn:

The DR-UPSI proposed in this paper is better than the comparison algorithm, and the experimental results also prove the effectiveness of the algorithm.

Compared with the ITCF based on traditional collaborative filtering and only simple linear fusion, the other algorithms based on deep learning significantly improve the recommendation effect, because the method based on deep learning can learn the nonlinear interaction between users and projects.

Compared with ITCF and DMF without user social relationship, other algorithms using social relationship improve the effect of recommendation. This is because users' social relationships, as auxiliary information, can greatly enrich users' information, greatly alleviate the problems of data sparsity and cold start, and then improve the recommendation effect.

Compared with CNSR and ScAN using users' social relations, DR-UPSI proposed in this paper comprehensively considers users' long-term and short-term preferences and users' preferences based on social influence, combines them into a complete user representation, and can effectively automatically extract user information and influence from friends when users interact with different projects, so as to learn better user representation and get more accurate recommendation results. From the experimental results, the DR-UPSI method is effective on CiaoDvD dataset HR@10 The index is 16.6% higher than CNSR and 6.5% higher than ScAN. Of DR-UPSI method NDCG@10 The index is 8.3% higher than CNSR and 6.9% higher than ScAN.

In order to further understand the importance of each module in the model, relevant ablation experiments were carried out in this paper. Compare the complete model with various variants to HR@10 and NDCG@10 For example. Among them, DR-UPSI represents the complete method in this paper, DR-UPSI_GRU represents the method of using traditional GRU. DR-UP means that the social impact of users is not considered DR-UPSI_GRU refers to the method of using traditional GRU without considering the social impact of users. Table 2 and Table 3 show the comparison results between DR-UPSI and various variants on the two datasets.

Table 2: Comparison results on CiaoDVD dataset

	HR@10	NDCG@10
DR-UP_GRU	0.3273	0.2176
DR-UP	0.3957	0.2943
DR-UPSI_GRU	0.4638	0.3125
DR-UPSI	**0.4913**	**0.3458**

Table 3: Comparison results on Douban dataset

	HR@10	NDCG@10
DR-UP_GRU	0.2203	0.1902
DR-UP	0.2736	0.2139
DR-UPSI_GRU	0.3031	0.2373
DR-UPSI	**0.3709**	**0.2514**

It can be seen from the results in the table that DR-UPSI_GRU has the worst effect, DR-UPSI_GRU is better than DR-UP, and DR-UPSI has the best effect. This is because DR-UPSI_GRU only uses user scoring sequence data, while DR-UP further considers the time interval between users' scoring behavior before and after, but does not consider the social impact of users. DR-UPSI_GRU, as the best variant method, considers both the user's scoring sequence and the user's social impact. However, the time interval between user scores, which can represent user preferences, is not considered. DR-UPSI proposed in this paper performs best because it comprehensively considers two characteristics that can affect user preferences. The first is to capture the changes of users' preferences in time through the improved Time-GRU. The second is to learn the impact of users' different friends on different projects through the user correlation matrix.

5. CONCLUSION

This paper proposes a dynamic recommendation algorithm because the existing social recommendation algorithms do not fully

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

consider the preference correlation between users and different friends for different items. According to the user scoring time data, the algorithm uses an improved gating cycle unit to process the user scoring data in continuous time to learn the user's long-term and short-term preferences. The user correlation matrix is proposed on the social impact of users to learn the expression of the impact of different friends on user preferences on different items. Finally, experiments on two real data sets show that the proposed algorithm has good results. In fact, there are many factors that affect users' preferences in social recommendation, such as the geographical location between users, the popularity of projects and the lack of explanation of recommendation results. These are the next research work.

REFRENCES:

[1] Liu HF, Jing LP, Yu J. Survey of matrix factorizationbased recommendation methods by integrating social information[J]. Ruan Jian XueBao/Journal of Software, 2018,29(2):340−362.
[2] Yang B, Lei Y, Liu J, et al. Social collaborative filtering by trust[J]. IEEE transactions onpattern analysis and machine intelligence, 2016, 39(8): 1633-1647.
[3] Peng W, Xin B. A social trust and preference segmentation-based matrix factorization recommendation algorithm[J]. EURASIP Journal on Wireless Communications and Networking,2019, 2019(1): 1-12.
[4] Guo G, Zhang J, Yorke-Smith N. TrustSVD: Collabor-ative Filtering with Both the Explicitand Implicit Influence of User Trust and of Item Ratings[C]//Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015: 123-129.
[5] Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//Proceedings of the 14th ACM SIGKDDinternational conference on Knowledge discovery and data mining. 2008: 426-434.
[6] Yan S, Lin K J, Zheng X, et al. An approach for building efficient and accurate social recommender systems using individual relationship networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2017,29(10): 2086-2099.
[7] Ye Xi-jun, Yuan Pei-sen, Gou Xiaoqing, et al. Collaborative filtering recommendation algorithm based on user interest and project cycle [J]. Journal of Nanjing University of Science and Technology,2018.42(4):392-400.
[8] Xue H J, Dai X, Zhang J, et al. Deep Matrix Factorization Models for Recommender Systems[C]//IJCAI. 2017, 17: 3203-3209.
[9] Fan W, Li Q, Cheng M. Deep modeling of social relations for recommendation[C]//32nd AAAI Conferenceon Artificial Intelligence, AAAI 2018. AAAI press, 2018: 8075-8076.
[10] Fan W, Ma Y, Yin D, et al. Deep social collaborativefiltering[C]//Proceedings of the 13th ACM Conferenceon Recommender Systems. 2019: 305-313.
[11] Pan Y, He F, Yu H. Learning social representations with deep autoencoder for recommender system[J]. World Wide Web, 2020, 23(4): 2259-2279.
[12] Chen J, Wang C, Shi Q, et al. Social recommendation based on users' attention and preference[J]. Neurocomputing, 2019, 341: 1-9.
[13] HE Ming, SUN Wang, XIAO Run, et al. Collaborative Filtering Recommendation Algorithm Combining Clustering and User Preferences[J]. Computer Science,2017,44(Z11):391-396.
[14] Qiu H, Guo G, Zhang J, et al. TBPR: Trinity preference based bayesian personalized ranking for multivariate implicit feedback[C]//Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization. 2016:305-306.
[15] Qiu H, Liu Y, Guo G, et al. BPRH: Bayesian personalized ranking for heterogeneous implicit feedback[J]. Information Sciences, 2018,453: 80-98.
[16] Le Wu, Sun P, Hong R, et al. Collaborative Neural Social Recommendation[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems,2021.51(1):464-476.
[17] Wang M, Ma J. A novel recommendation approach based on users' weighted trust relations and the rating similarities[J]. Soft Computing, 2016, 20(10): 3981-3990.
[18] Pal B, Jenamani M. Trust inference using implicit influence and projected user network for item recommendation[J]. Journal of Intelligent Information Systems, 2019, 52(2): 425-450.
[19] Li M, Tei K, Fukazawa Y. An efficient co-attention neural network for social recommendation[C]//2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI). IEEE,2019: 34-42.
[20] CHAI Rui-min, YIN Chen, MENG Xiang-fu, et al. A recurrent neural network model basedon spatial and temporal information for next POI recommendation[J]. CAAI transactions onintelligent systems,2020:1-10.