U.P.B. Sci. Bull., Series C, Vol. 84, Iss. 2, 2022 ISSN 2286-3540

A HYBRID RECOMMENDATION ALGORITHM BASED ON USER CHARACTERISTICS

Hong-zhi YU [1], Deng-yun ZHU[1] , Fu-cheng WAN [1*,2], Tian-tian WU[1], Ma NING[1]

Shop recommendation system is an important part of the e-commerce recommendation system. Shop recommendation system in this paper mainly consist of three steps. First, constructed matrix decomposition module and deep network module for tackling scores data and comment data, and then connected two modules by weight factor, trained by the same loss function, at last the comprehensive score is output by scoring prediction, analyzing fusion factor for the effect of algorithm through the preprocessed text and the parameter setup fusion model. Each experiment adopts the five-fold crossover verification method, and the prediction accuracy of this algorithm compared with other five different algorithms. Experimental results verify that the UFFSR algorithm can effectively improve the accuracy of prediction scoring and alleviate the data sparsity and cold start problems to a certain extent.

Keywords: Recommender System, Rating Prediction, User Characteristic, Deep Learning

1 Introduction

With the rapid development and wide application of Internet and big data technology, a large amount of data is formed in various fields all the time. It is extremely difficult to quickly mine the information they need from massive data because different people are interested in different content. Therefore, it has become an urgent information overload problem to be solved on, and the emergence of recommendation system provides unprecedented convenience for many users. The first research on the recommendation system began in developed countries such as the United Kingdom and the United States. Researchers began to carry out in-depth research on recommendation algorithms and applications, which have proposed and improved a variety of recommendation algorithms and applications. Recommendation algorithms mainly include traditional recommendation methods and deep learning recommendation methods.

[1] Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou 730030, China;
2 Key Laboratory of China's Ethnic Languages and Intelligent Processing of Gansu Province, Northwest Minzu University, Lanzhou 730030, China;
* Corresponding author: 306261663@qq.com

Deep learning technology has made rapid progress in many fields, and it also provides new research ideas and directions for recommendation system, which has quickly become a research hotspot in recommendation system.

Cheng and other scholars put forward Wide & Deep deep learning model, which combines traditional linear model and neural network to deal with low-order and high-order features respectively [1]. Covington et al. tried deep neural networks (DNN) in YouTube video recommendation system, and put forward a DNN model, which was later introduced into its own recommendation system by Tencent and other large Internet companies [2]. Gong and other scholars put forward a Convolutional Neural Networks (CNNs) model based on attention mechanism, which is used to recommend topic tags in Weibo. The research data show that this model is superior to the most advanced method of the same category at that time [3]. Kim et al. put forward a new context-aware model-Convolutional Matrix Factorization (ConvMF) model. The principle is to filter and capture the context information in the text by combining convolution neural network with matrix factorization technology, which improves the rating prediction accuracy [4]. Karatzoglou and other scholars have analyzed the deep learning technologies widely used in the current recommendation system, such as cyclic neural network and convolutional network, etc. The deep learning is gradually popularized in the recommendation system [5]. Zheng and other scholars put forward a Deep Cooperative Neural Networks, DeepCoNN). The principle is to process comment texts through convolution neural networks, model users and projects respectively, and achieve the purpose of rating prediction. Experiments on multiple data sets show that the prediction results are better than most current benchmark recommendation methods [6]. Chen and other scholars put forward a NARRE (Neural Attentional Regression model with Review-level Explanations) model, which focuses on the validity of comment information on e-commerce platform and gives weight to each comment by adopting Attention mechanism to improve the rating prediction accuracy [7]. Wu et al. proposed to use Graph Neural Networks (GNN) to make session-based recommendation [8], regard users and projects as nodes, and model users by GNN to mine more accurate user behavior features.

Although deep learning shows unique advantages in the field of recommendation, there are also some problems, which need further optimization and improvement [9].

This paper examines the following main aspects:

Shop recommendation algorithm UFFSR, which combines user characteristics, is improved and designed, and the algorithm consists of two sub-modules, namely, matrix decomposition module and deep network module, which process the user's scoring matrix through the hidden factor model, and then uses the two-channel neural network to mine the user's personal preference information and the store's attribute characteristics from the comments, and finally establishes the association between the shop and the user. By analyzing the effect of different number of hidden factors on the RMSE value and MAE value of the algorithm evaluation index, the optimal RMSE value and the number of hidden factors of MAE value were obtained by the UFFSR algorithm on the Yelp data set [10], and the influence of fusion factor on the algorithm was also studied.

2. Model Design

Features of users are usually a single attribute or a combination of attributes of users. From the perspective of source data, this paper divides user features into two categories: rating user features and commenting user features. Rating user features refer to features that learned from rating information and commenting user features refer to features that learned from commenting information [11]. For data sets without comment information in the early stage, collaborative algorithm or decomposition class algorithm is usually used to learn. At this time, no matter what features are learned, they are further expressions of user interaction rating behavior, so they are called rating user features. For data sets containing comment texts, we can use natural language processing (NLP) to further mine user features from comment texts, which we call comment user features [12].

2.1 Algorithm framework design

According to the definition of two kinds of user features, based on decomposition class recommendation algorithm and comment text recommendation algorithm [13], this paper proposes a framework of Rating and Review User Feature Fusion for Recommendation (RRUFFR). The fusion model framework flow is shown in Fig. 1.

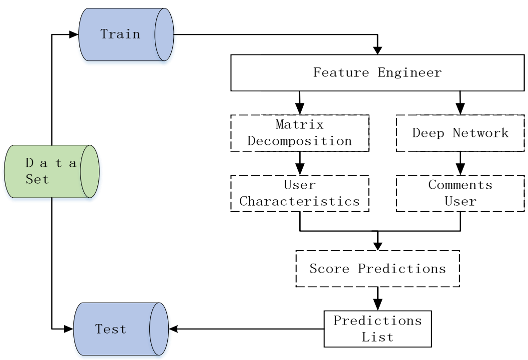

Fig. 1. Framework diagram of fusion model

First, ratio of training data set to test data set is set to be 0.8: 0.2, and the whole process is randomly segmented. Next, feature engineering is carried out, including length normalization, deletion of stop words and calculation of TF*IDF value. User feature vectors and item feature vectors based on comment text are obtained by using pre-trained GoogleNews-vectors-negative300. bin; The other part processes rating data, inputting the rating data into a matrix decomposition module to obtain rating user features based on rating information, and inputting user feature vectors and item feature vectors into a depth network module to obtain comment user features based on comment text; Then the two modules are connected by weight factor and trained by the same loss function; Finally, the recommendation accuracy of the fusion model on test set is obtained by rating prediction and output rating [14].

2.2 Model

Based on the RRUFFR, the existing recommendation algorithms are improved, and a shop UFFSR is designed. The specific ideas are as follows:

The matrix decomposition module adopts hidden factor model, and the depth network module adopts improved convolution neural network model, which are fused to form a fused hidden factor model and a rating prediction recommendation model based on convolution neural network [15]. The hidden factor model part is carried out according to direct rating information, and matrix learning is carried out to realize predicted rating; The dual-channel convolution neural network includes two parallel convolution neural networks: user network (Netu) and project network (Neti). Netu pays more attention to users, mining users' personal preferences and interests according to comments published by users, while Neti pays more attention to commodities, learning the features of commodities according to comments on commodities. Finally, the convolution neural network model establishes the association between users and items by means of factorization and realizes the predicted rating R_2. Finally, R_1 and R_2 are added by weighted method to realize comprehensive predicted rating. Fig. 2 shows the overall framework of the model.

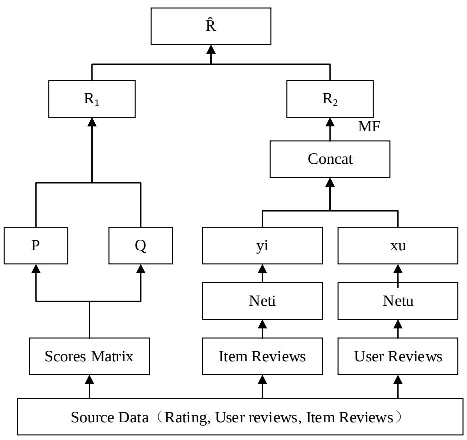

Fig. 2 Model architecture diagram

Because the user network Netu and the project network Neti have the same structure, the word embedding layer is the first step of the model, and the comment information of users and items is mapped into a word vector matrix as the input of convolutional layer; The second and third layers are used as neural network layers to mine the feature attributes of users and items in abstract space [16]; The fourth layer is the full connection layer of the model. Take Netu for example to introduce the implementation details between networks in detail, as shown in Fig. 3.

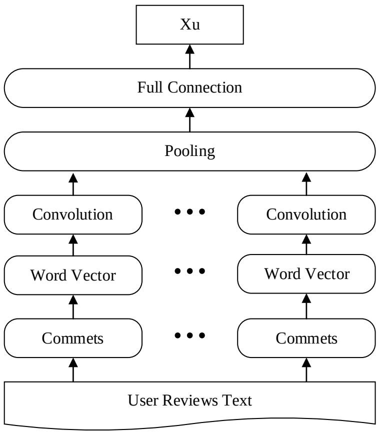

Fig. 3 User Network Netu Architecture

2.2.1 Word vector layer

Word vector mapping relationship $f : M \to R^n$, which maps n-dimensional vectors in dictionary M. The value of n is 300. In the word embedding layer, the comment text will be mapped to the word vector matrix as important semantic information, which is convenient for mining its feature attributes. The specific mathematical expression process is to integrate the user u's comment word vector and take it as a document set D, and the length of D contains N words, then the user u's comment word vector matrix M is:

$$M = \theta(d_1) \oplus \theta(d_2) \oplus \theta(d_3) \oplus \cdots \oplus \theta(d_n) \quad (1)$$

Where: d_k represents the k-th word in document d, k=1, 2, *, n; Then $\theta(d_k)$ is mapped as a word d_k to the representation of the corresponding n-dimensional word vector space.

2.2.2 Convolutional layer

The convolutional layer is mainly composed of many convolution filters with different windows, which is used to mine the user comment vector matrix M_u and extract its abstract features. The output of word embedding layer is used as its input matrix, and the window of convolution filter is usually much smaller than the size of the input matrix, so local information of the input matrix can be extracted. Set the number of neurons be n, in which neuron j uses convolutional kernel $K_j \in R^{c \times t}$ with word window size t. Therefore, for M_u, the convolutional result of each convolutional kernel K_j can be expressed as follows:

$$M = f\left(M_u * K_j + b_j\right) \tag{2}$$

Where the symbol "*" indicates convolution operation, b_j is offset term, f is activation function and modified linear activation unit, and the expression is:

$$f\left(x\right) = \max\left(0, x\right) \tag{3}$$

2.2.3 Pooling layer

Generally speaking, the methods of data processing in pooling layer include mean method, sum method, maximum pooling and pooling. For the model architecture, it is necessary to eliminate the noise features contained in the feature map vector obtained from the convolutional layer, so it is necessary to choose a reasonable pooling operation method. For comment text information, if it is pooled by summation or mean, noise will be introduced, which will have adverse effects on the further modeling of text information. However, if it is pooled by extracting the maximum value, only the maximum value of feature map vector will be extracted as the subsequent input, which meets the modeling requirements. Therefore, choosing the maximum pooling method, one-dimensional MaxPooling, will reduce the output k_i to a vector of constant size by maximum pooling layer convolution:

$$o_j = max\left\{k_1, k_2, k_3, \cdots, k_{(n-t+1)}\right\} \tag{4}$$

The above equation represents the operation result of a convolutional kernel. In order to extract many different features, the model uses k convolutional kernels, and the output vector is expressed as follows:

$$O = \left(o_1, o_2, o_3, \cdots, o_k\right) \tag{5}$$

2.2.4 Full connection layer

After passing through the second convolutional layer, the data processing result enters the full connection layer, and finally obtains a high-level abstract feature vector about the user u, which is expressed as $F_u \in R^{d \times 1}$. The following is the mathematical expression:

$$F_u = f\left(W \cdot O + b^{`}\right) \tag{6}$$

In the equation, the weight parameters of the full connection layer are expressed as W matrix; The offset item is $b^{`}$. Finally, the output x_u of user convolutional network Netu and the output y_i of item convolutional network Neti can be obtained.

2.2.5 Shared layer

In order to accurately predict the user's rating of items, it is necessary to map the user and item features x_u and y_i in different feature spaces into the same feature space. Then, you need to splice x_u with y_i to get $\hat{z}$. The specific expression is as follows:

$$\hat{z} = x_u \oplus y_i \tag{7}$$

Where $\oplus$ is the Concat operation.

The obtained vector not only retains the hidden feature information of users and items, but also contains the intersection information of them to a certain extent. Therefore, in order to further obtain the interaction, the factorization machine model is selected as the rating prediction function, and the specific equation is as follows:

$$R_2 = w_0 + \sum_{i=1}^{|\hat{z}|} w_i \hat{z}_i + \sum_{i=1}^{|\hat{z}|} \left\langle v_i | v_j \right\rangle \sum_{j=i+1}^{|\hat{z}|} \hat{z}_i \hat{z}_j \tag{8}$$

$$V = \begin{bmatrix} v_{11} & \cdots & v_{1k} \\ \vdots & \ddots & \vdots \\ v_{|\hat{z}|1} & \cdots & v_{|\hat{z}|k} \end{bmatrix} = \begin{pmatrix} v_1 \\ \vdots \\ v_{|\hat{z}|} \end{pmatrix} \tag{9}$$

Among them, $|\hat{z}|$ is the latitude of the feature vector $\hat{z}$; $w_0 \in R$ is the global deviation; $w_i \in R$ is the weight of the i-th variable in the feature vector; $V \in R^{|\hat{z}| \times k}$ is a coefficient matrix; $\langle v_i | v_j \rangle$ is the dot product of the i-dimensional and j-dimensional vectors of sparse matrix.

2.3 Model training

Training process of network model can be divided into two steps: forward propagation and backward feedback, which are described as follows.

Stage 1: Forward propagation stage.

(1) In order to speed up the convergence of the network model, this paper pre-trains the vector representation of vocabulary and comments, and then inputs the pre-trained initial vectors into the network.

(2) In order to ensure the consistency of the distribution variance of the input and output data of each layer parameters in the network, the Xavier method is adopted to initialize the network weights. The variance of parameters obeys the following uniform distribution:

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_i+n_{i+1}}}, \frac{\sqrt{6}}{\sqrt{n_i+n_{i+1}}}\right] \tag{10}$$

(3) To mitigate the gradient dissipation, the modified linear activation unit (ReLUs) activation function is used in the convolutional layer and the fully connected layer. In the process of network forward propagation, data is transmitted from front to back, and the abstract expression of features is learned until the final output layer, and the user's predicted rating R_2 of goods is output.

Stage 2: Back propagation process.

(1) Loss function. To minimize the difference between the predicted rating $\hat{R}$ and the actual rating R, it is expressed as follows:

$$\begin{aligned} Loss &= \hat{R} - R = a*(R_1 - R) + (1-a)*(R_2 - R) \\ &= a*\left\{\sum_{(u,i)\in K}\left(r_{ui} - \sum_{1}^{f} p_{u,f}q_{i,f}\right)^2 + \lambda\left\|p_u\right\|^2 + \lambda\left\|q_i\right\|^2\right\} + \\ &\quad (1-a)*\left\{w_0 + \sum_{i=1}^{|\hat{z}|} w_i\hat{z}_i + \sum_{i=1}^{|\hat{z}|}\langle v_i | v_j\rangle \sum_{j=i+1}^{|\hat{z}|}\hat{z}_i\hat{z}_j\right\} \end{aligned} \tag{11}$$

Among them: r_{ui} is the actual rating of item i for the user u.

(2) Parameter learning process. The main parameters in this fusion model are P matrix, Q matrix, neuron weight and bias, which together determine the accuracy of the model output. By solving the gradient of the objective function, the prediction error can be propagated forward from the last layer, and the network parameters can be updated layer by layer. Finally, the training of the model is completed iteratively by forward propagation and backward feedback operations.

3. Experiment

3.1 Experimental Corpus

In order to provide users with accurate personalized shop recommendation service and fully evaluate the effectiveness of the improved UFFSR algorithm, our experiment uses Yelp public data set to verify the performance of the algorithm:

(1) Yelp dataset is a large dataset about review websites. Covering shops, reviews and user data, It contains six json files, they are business.json (transaction table), user.json (user table), review.json (comment table), checkin.json (sign-in table), tip.json (suggestion table) and photo.json (picture table). There is Multidimensional data in the transaction table, including store name, evaluation quantity, star rating, geographical location, business hours and other important factors that users pay more attention to, such as booking method, whether to take it out, WiFi, parking lot and other store attributes. The comment form contains key information about star rating, evaluation preference, evaluation praise and evaluation time. By filtering the transaction table data and retaining the relevant data of shop evaluation, the evaluation texts obtained from shops constitute the corpus of the project, which contains about one million comments and rating information.

Data set contains statistics such as the number of shops (Items), the number of users (Users), the number of reviews (Reviews), the mean number of reviews (Review Per User), the mean number of words (Words Per Review) and the density of the data set.

$$Density = \frac{n(ratings)}{n(users) \times n(items)} \tag{12}$$

$$Review\,Per\,User = \frac{n(words)}{n(users)} \tag{13}$$

$$Words\,Per\,Review = \frac{n(words)}{n(reviews)} \tag{14}$$

Among them, $n(ratings)$ is the number of rating, $n(users)$ is the number of users and $n(items)$ is the number of shops, and $n(words)$ means the sum of all words. In the experiment, the data set is divided into training set, verification set and test set according to the ratio of 8: 1: 1. Five experiments are carried out on the data set, and five experiments are random from training, verification, test data segmentation to the end of test process, which can fully ensure the randomness of each experiment and improve the correctness of the experimental results.

3.2 Experimental comparison

In order to evaluate the application effect of the algorithm in shop recommendation, the accuracy of rating prediction task, the mean absolute error

(MAE), mean square error (MSE) and root mean square error (RMSE) can be selected as evaluation indexes to judge the accuracy of prediction results.

(1) Mean Absolute Error (MAE)

$$MAE = \frac{\sum_{r_{ij} \in T} \left| r_{ij} - \hat{r}_{ij} \right|}{|T|} \tag{15}$$

(2) Mean Square Error (MSE)

$$MSE = \frac{1}{|T|} \sum_{r_{ij} \in T} \left(r_{ij} - \hat{r}_{ij} \right)^2 \tag{16}$$

(3) Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{r_{ij} \in T} \left(r_{ij} - \hat{r}_{ij} \right)^2} \tag{17}$$

Among them, r_{ij} is the real rating of user i on item j in the data set, $\hat{r}_{ij}$ is the predicted rating of user i on item j in the experiment, and T is the total number of user rating in test set. Among them, the smaller the value of MAE, MSE and RMSE evaluation index, the smaller the difference between the predicted rating and the actual rating, and the higher the prediction accuracy.

3.3 Parameter setting

In the aspect of experimental parameter setting, we mainly refer to the experience of parameter setting accumulated by other researchers in the experimental process, as follows:

(1) Parameter setting of the benchmark algorithm.

For SVD and SVD + + algorithms, the default value of learning rate is 0.007, the default value of regularization parameters is 0.02, the default number of hidden factors is 20, and the subsequent setting are [10, 30, 40, 50, 60] for comparative experiments, and the iteration rounds are 50 rounds.

For NMF algorithm, the default value of regularization term λ_u and λ_i is 0.06, the default value of regularization term b_u and b_i is 0.02, the learning rate is set to 0.005, the number of hidden factors is set to 15 by default, and the subsequent hyperparametric analysis is set to [10, 20, 30, 40, 50, 60], and the iteration rounds are 50 rounds.

For DeepCoNN algorithm, the word embedding dimension is set to 300, the learning rate is set to 0.002, the Dropout is set to 0.5, the number of convolution filters is set to 100, the size is set to 3, the number of samples per batch is set to 32 by default, the subsequent comparison is set to [64, 128, 256], and the iteration rounds are set to 10 rounds.

For NARRE algorithm, the word embedding dimension is set to 300, the learning rate is set to 0.002, the Dropout is set to 0.5, the number of convolution filters is set to 100, the size is set to 3, the number of samples per batch is set to 32 by default, the subsequent comparison is set to [64, 128, 256], and the iteration rounds are set to 10 rounds.

(2) The improved algorithm parameter setting in this paper.

Because it is a fusion model, the parameters of matrix decomposition module refer to LFM algorithm, and the parameters of neural network module refer to DeepCoNN algorithm. The weight factor defaults to 0.2, and the subsequent values are [0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] for comparative experiments.

3.4 Comparison with previous experiments

For the convenience of analysis, the comparison algorithms SVD, SVD + + and NMF are collectively referred to as the first kind of algorithm, and the comparison algorithms DeepCoNN and NARRE based on user comment text are collectively referred to as the second kind of algorithm. Observed and analyzed from the table, the following conclusions are obtained:

(1) Yelp dataset, the improved UFFSR algorithm outperforms other comparison algorithms, and its RMSE and MAE values are lower than other comparison algorithms. The overall performance of the algorithm is UFFSR > NARRE > DeepCoNN > SVD > SVD + + > NMF. The results show that the new model can improve the prediction accuracy and the rating prediction effect of recommendation algorithm, users can make more accurate recommendations.

(2) By comparing the UFFSR algorithm and the second class algorithm (DeepCoNN, NARRE) using comment text and rating with the first class algorithm (NMF, SVD, SVD + +) using rating only, the experimental results show that the model based on comment text has more advantages in learning finer-grained information on Yelp data set, which is more conducive to improving the rating prediction effect.

(3) By comparing the improved UFFSR algorithm studied in this paper with the second kind of algorithm (DeepCoNN, NARRE), Compared with DeepCoNN, which does not train the overall rating information as the source data, the fusion algorithm of matrix decomposition with sub-modules has better recommendation accuracy. On the premise of keeping the neural network consistent, from table 1, it can be seen that the addition of matrix decomposition sub-modules really improves the prediction accuracy of the overall model.

Table 1

Algorithm performance analysis

Algorithm	RMSE Performance	MAE index
SVD	1.3761	1.1403
SVD + +	1.3803	1.1459
NMF	1.5401	1.2742
DeepCoNN	1.1543	0.9113
NARRE	1.1027	0.8612
UFFSR	**1.0215**	**0.7835**

For UFFSR algorithm, the evaluation indexes RMSE and MAE are lower than the other five algorithms, especially compared with the first kind of algorithms (NMF, SVD, SVD + +), because the smaller the evaluation indexes RMSE and MAE, the smaller the difference between the predicted rating and the real rating, and the higher the prediction accuracy. Therefore, through comparative experiments, the UFFSR algorithm in this paper can effectively improve the accuracy of recommendation.

3.5 Parameter tuning analysis of hidden factor model

For matrix model, the number of hidden classes f is an important super parameter. Therefore, it is necessary to analyze the influence of the number of hidden classes on the results of the algorithm involving the number of hidden classes in this experiment.

The number of hidden classes f is set to 8, 16, 32, 64, and other parameters remain unchanged. The experimental results are shown in Table 2 and Table 3.

Table 2

Comparison of RMSE under different factors

Factors	SVD	SVD + +	NMF	DeepCoNN	NARRE	**UFFSR**
8	1.3761	1.3754	1.6006	1.1523	1.1027	**0.9189**
16	1.3767	1.3787	1.5511	1.1498	1.1083	**0.9107**
32	1.3765	1.3839	1.5269	1.1562	1.1134	**0.9105**
64	1.3784	1.3933	1.5339	1.1341	1.0852	**0.9118**

It is easy to observe that on Yelp datasets, the performance of NMF algorithm is most sensitive to the fluctuation of the number of hidden factors, especially when the number of hidden factors rises from 16 to 32. On Yelp dataset, when the number of hidden factors is set to 32, UFFSR algorithm will get the best RMSE value.

Table 3

Comparison of MAE under different factors

Factors	SVD	SVD + +	NMF	DeepCoNN	NARRE	UFFSR
8	1.1403	1.1377	1.3498	0.9079	0.8437	0.6553
16	1.1410	1.1435	1.2899	0.9046	0.8218	0.6567
32	1.1412	1.1516	1.2508	0.9129	0.8372	0.6563
64	1.1438	1.1642	1.2384	0.8841	0.8491	0.6488

On Yelp dataset, the variation of the number of hidden factors in SVD algorithm has little fluctuation on recommendation performance, while NMF algorithm is sensitive to fluctuation with the increase of the number of hidden factors, especially when the number of hidden factors rises from 16 to 32, the change of the number of hidden factors of other algorithms has little fluctuation on the recommendation performance. On Yelp dataset, when the number of hidden factors is set to about 64, UFFSR algorithm will get the best MAE value.

3.6 Influence of fusion factor on algorithm results

This paper focuses on the fusion model of rating and commenting user features. It can be intuitively felt through the calculation equation of comprehensive predicted rating 3-11, so it is very important to study the influence of fusion factor a on experimental results. From the display, the fusion factor a controls the weight of the rating prediction of the hidden factor model, and (1-a) controls the weight of the partial rating prediction of the convolution neural network. A is a super parameter. For different data sets, the final best effect of the model is not consistent. The comparison results of MAE and RMSE with different weights on Yelp data sets are shown in Fig. 4 and Fig. 5.

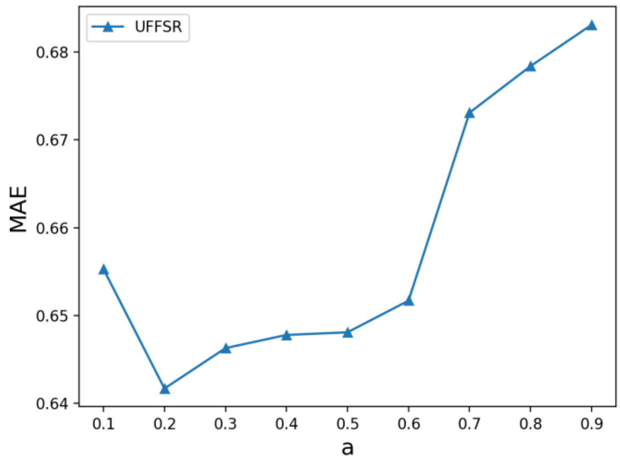

Fig. 4 Comparison of MAE with different weights on Yelp data set

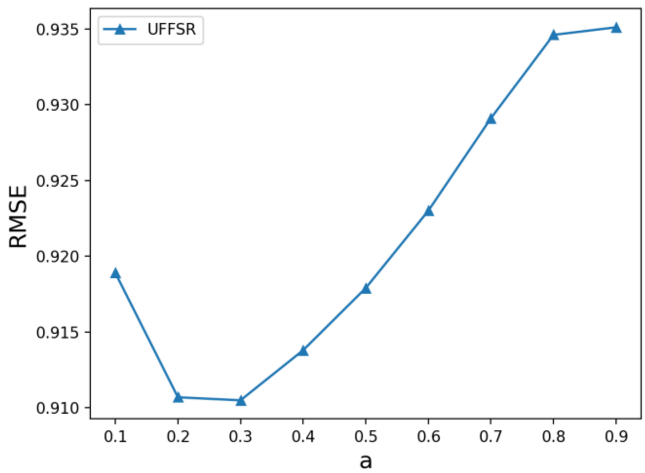

Fig. 5 Comparison of RMSE with different weights on Yelp data set

It shows that the fusion factor ***a*** controls the weight of the rating prediction of the hidden factor model, that is, the weight of the matrix decomposition module. According to the results of Yelp data set, when the weight factors a is within the range of [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] and the weight of the fusion model is in the range of 0.2 ~ 0.3, the overall prediction accuracy is better. Compared with the depth network module, the hidden factor module contributes more "loss" to the overall loss function, so it is necessary to make the weight factor a take a smaller value. That is, in the rating prediction task, the smaller the "contribution" of the hidden factor module to the results, the better the overall results of the model can be achieved. Rating information is regarded as the overall evaluation of shops by users, while comment text is a finer-grained evaluation of shops by users. From the weight factor analysis, local model has greater influence on rating prediction than global model.

4. Conclusion

Five experiments are used to compare the algorithms. In the same data set and experimental environment, the UFFSR is implemented in code. Each experiment uses a 50-fold cross-validation method to verify and analyze all the comparison algorithms and the UFFSR algorithm. The results show that the improved UFFSR algorithm is better than the other five comparison algorithms on Yelp data set. In order to prove the effectiveness of two sub-modules of fusion algorithm UFFSR, the effectiveness of matrix decomposition module is verified by comparing with SVD, SVD + + and NMF, and the effectiveness of deep network model is verified by comparing with DeepCoNN and NARRE. Finally, the influence of weight factors on experimental results is analyzed and demonstrated.

In the next step, the multi-dimensional features such as occupation, age and gender of users will be integrated to explore the influence of different features on the recommendation model.

Acknowledgements

This work was supported by talent innovation and entrepreneurship project of Lanzhou Chengguan District Science and Technology Bureau(2021RCCX0016).

R E F E R E N C E S

[1]. *Jie Zhou, Wei Xu*. End-to-end learning of semantic role labeling using recurrent neural networks //Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 2015: 1127-1137.

[2]. *Gong Y, Zhang Q.* Hashtag recommendation using attention-based convolutional neural network//IJCAI. 2016: 2782-2788.

[3]. *Kim D, Park C, Ob J, et al.* Convolutional matrix factorization for document context-aware recommendation//Proceedings of the 10th ACM Conference on Recommender Systems. 2016: 233-240.

[4]. *Karatzoglou A, Hidasi B.* Deep Learning for Recommender Systems. in: Proceedings of the Eleventh ACM Conference on Recommender Systems. Como, Italy: ACM, 2017: 396-397.

[5]. *Koren Y, Bell R, Volinsky C.* Matrix Factorization Techniques for Recommender Systems. Computer, 2009, 42(8): 30-37.

[6]. *Burke R.* Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-adapted Interaction, 2002, 12(4): 331-370.

[7] *LeCun Y, Bengio Y, Hinton G.* Deep learning. Nature, 2015, 521(7553): 436-444

[8] *Silver D, Huang A, Maddison C J, et al.* Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529(7587): 484-489.

[9] *Mikolov T, Karafiát M, Burget L, et al.* Recurrent neural network-based language model//Eleventh annual conference of the international speech communication association. 2010.

[10] https://www.yelp.com/dataset

[11] *Sutskever I, Vinyals O, Le Q V.* Sequence to sequence learning with neural networks//Advances in neural information processing systems. 2014: 3104-3112.

[12] *Sak H, Senior A, Beaufays F.* Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128, 2014.

[13] *Adomavicius G, Tuzhilin A.* Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6):734-749

[14] *Goldberg D, Nichols D, Oki B M, et al.* Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, 1992, 35(12):61-70.

[15] *Resnick P, Iacovou N, Suchak M, et al.* GroupLens: an Open Architecture for Collaborative Filtering of Netnews. in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. Chapel Hill, North Carolina, USA: ACM, 1994:175-186

[16] *Upendra Shardanand and Pattie Maes.* 1995. Social information filtering: algorithms for automating “word of mouth” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95). ACM Press/Addison-Wesley Publishing Co., USA, 210–217. DOI:https://doi.org/10.1145/223904.223931.