

A NOVEL NEURAL COLLABORATIVE FILTERING
RECOMMENDATION BASED ON SIDE INFORMATION
FUSION

Ruihui Mu

Received on July 19, 2022
Presented by Ch. Roumenin, Member of BAS, on October 25, 2022

Abstract

It is difficult to accurately learn user's latent features using only one single data source. In order to solve these problems, we consider to utilize relevant side information of users or items as a supplement to rating information to enhance the performance of recommender systems, and propose a novel neural collaborative filtering recommendation model based on side information fusion. Extensive experiments on different datasets validate the efficiency and accuracy of our proposed framework.

Key words: neural network, side information, denoising autoencoder, rating information

2020 Mathematics Subject Classification: 68T07

1. Introduction. With the rapid development of DL techniques, the application of DL techniques has attracted more and more attention recently and has become a hot topic in the field of recommender systems. Deep learning [1] can automatically extract low-level features from multi-source heterogeneous data, fuse them into dense and deep high-level features, and mine the hidden rules in the data, thus solving the problem of manual feature extraction in traditional recommender systems. On this basis, integrating with traditional recommendation algorithms can not only effectively utilize heterogeneous auxiliary information from multiple data sources, but also alleviate the problem of sparse user-item

historical interaction data and cold start in traditional algorithm models. Since the side information contains a wealth of relevant data information about users and items, it can effectively solve the problems existing in traditional algorithm models.

To solve this problem, we propose a novel neural collaborative filtering recommendation model based on side information fusion, which constructs a deep generation model based on stacked denoising autoencoder (SDAE) and fuses side information and user-item historical interaction information to mine user's latent preferences. The model includes user modelling and item modelling, which adopts the SDAE network structure, respectively. We also propose a tightly coupled correlation regularization term, which can learn the user's latent preference from user side information, item image information, and item description information, and the model structure is very flexible and can be easily extended to other types of side information. In order to verify the recommendation quality of the proposed model, we conduct multi-angle comparative experiments on two different datasets. Experiments show that the proposed model is superior to the existing typical recommendation algorithms, and can effectively solve the data sparseness and model scalability in traditional recommendation algorithms.

The remainder is organized as follows. We present our proposed framework completely in Section 2. Then, Section 3 conducts extensive experiments. Finally, Section 4 gives conclusion with future work.

2. Methodology. 2.1. Model overview. In this section, we elaborate our proposed framework, as shown in Fig.1. Our framework mainly includes three parts: user modelling part, item modelling part and feature fusion part. Among them, the user and item modelling part include five modules: user auxiliary information feature extraction module, user rating information feature extraction module, item rating information feature extraction module, item image information feature extraction module and item description information feature extraction module. In the process of feature extraction, we adopt SDAE to learn and extract the latent representation of auxiliary information separately, and then use the MLP module to fuse the latent features of user and item to further mine deep non-linear relationship of auxiliary information, and perform rating prediction and top- k list recommendation.

Specifically, we employ three techniques (SDAE, CSDAE, and DNN) to extract latent representations of user auxiliary information, user rating information, item rating information, item image information, and item description information, respectively, and integrate these latent representations into a unified model for training.

2.2. Feature extraction. 2.2.1. User latent feature modelling.

The user latent feature extraction module includes the latent feature extraction of user auxiliary information and the latent feature extraction of user rating information. We employ two SDAEs to extract features, respectively, and then

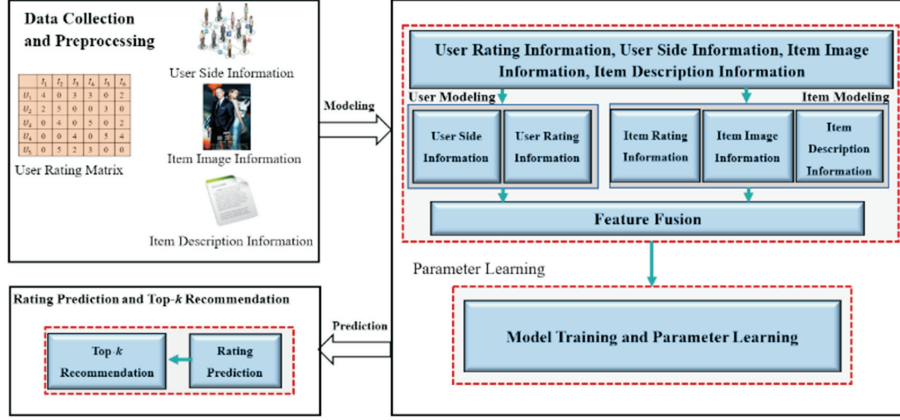


Fig. 1. The overview of framework

adopt multi-layer perceptron to fuse the user's latent features.

– *The latent feature extraction of user auxiliary information:* As shown in Fig. 1, we adopt SDAE to extract the latent feature from user auxiliary information. Assuming that the number of neural network layers is L_s , the encoder is the first $\frac{L_s}{2}$ layers and the decoder is the following $\frac{L_s}{2}$ layers. The encoder is responsible for mapping the corrupted input $\tilde{S}$ to latent representation $H_{L_s/2}$, the decoder is responsible for reconstructing the original user auxiliary information S from the latent representation $H_{L_s/2}$.

We adopt Equation (1) to obtain the latent representation of any hidden layer l and the output of L_s layer as follows:

$$\begin{aligned}
 (1) \quad & H_1 = g(W_1^s \tilde{S} + b_1^s) \\
 & H_l = g(W_l^s H_{l-1} + b_l^s) \\
 & \hat{S} = f(W_{L_s}^s H_{L_s-1} + b_{L_s}^s),
 \end{aligned}$$

where $l \in \{2, \dots, L_s - 1\}$, $g(\cdot)$ represents the mapping function of the hidden layer, and $f(\cdot)$ represents the function of the output layer, which is usually nonlinear. W_*^s represents the weight of each layer, and b_*^s represents the bias of each layer.

We adopt $H_s = H_{L_s/2}$ to represent the latent feature of the user auxiliary information, and $\hat{S}$ represents the output of the SDAE module, that is, the reconstruction of original user auxiliary information S . $\tilde{S}$ represents the corrupted user auxiliary information (some entries of the user auxiliary information are randomly set to zero by masking the noise method).

SDAE employs deep neural network to reconstruct its original input, minimizing the squared error between the input and its reconstruction. In user social relations, there is a phenomenon of social association, that is, the features of users (such as preferences and behaviours) are related to their social relations. At the

same time, since the social matrix is binary data, we choose the widely used log loss function for modelling, which can be regarded as the negative log-likelihood of the output of binary data with Bernoulli distribution. In this method, we select the cross-entropy loss function as the loss function of the Top- k recommendation task, which is formally defined as Equation (2), and use the back-propagation method to further adjust the bias and weight of each layer.

$$(2) \quad \mathcal{L}_s = \sum_{j=1}^M \left\{ \sum_{i=1}^M \left(-S_{i,j} \log(\hat{S}_{i,j}) - (1 - S_{i,j}) \log(1 - \hat{S}_{i,j}) \right) \right\} + \left(\sum_{j=1}^M \sum_{i=1}^M S_{i,j} \|H_{s_i} - H_{s_j}\|_2^2 \right) + \frac{\lambda_1}{2} \left(\sum \|W_l^s\|_2^2 + \sum \|b_l^s\|_2^2 \right).$$

Here λ_1 represents the regularization coefficient to prevent overfitting, W_*^s represents the weight matrix, and b_*^s represents the bias vector. H_{s_i} and H_{s_j} represent the latent features of the social relationship of user u_i and user u_j , respectively, that is, the middle-hidden layer of SDAE. The first term represents the reconstruction error and the second term represents the regularization term between latent features of users' social relations. Here $\|\cdot\|_2^2$ is L2-norm regularization, which is given by the square root of the sum of the squares of its elements. Specifically, if user u_i has a social relationship with user u_j ($S_{i,j} = 1$), then user u_i and user u_j have similar interests or are influenced by user u_j , i.e., they have similar behaviours. We place a social correlation regularizer to minimize the difference in social latent vectors between them.

– *The latent feature extraction of user rating information:* The latent feature extraction of user rating information is similar to that of user auxiliary information. We also use SDAE network for latent feature extraction. Assuming that the number of neural network layers is L_r , the encoder is the first $\frac{L_r}{2}$ layers and the decoder is the following $\frac{L_r}{2}$ layers. The encoder is responsible for mapping the corrupted input R'' to latent representation $H_{L_r/2}$, the decoder is responsible for reconstructing the original user rating information R_u from the latent representation $H_{L_r/2}$.

We employ Equation (3) to obtain the latent representation of any layer l and the output of L_r layer as follows:

$$(3) \quad \begin{aligned} H_1 &= g(W_1^r \tilde{R} + b_1^r) \\ H_l &= g(W_l^r H_{r,l-1} + b_l^r) \\ \hat{R}_u &= f(W_{L_r}^r H_{L_r-1} + b_{L_r}^r), \end{aligned}$$

where $l \in \{2, \dots, L_r - 1\}$, $g(\cdot)$ represents the mapping function of the hidden layer, and $f(\cdot)$ represents the function of the output layer, which is usually nonlinear. W_*^r represents the weight of each layer, and b_*^r represents the bias of each layer.

We use $H_u = H_{L_r/2}$ to represent the latent feature of the user rating information, and $\hat{R}_u$ to represent the output of the SDAE module, i.e., the reconstruction of original user rating information R . $\tilde{R}$ represents the corrupted user rating information.

The objective function is defined as Equation (4):

$$(4) \quad \mathcal{L}_{R_u} = \sum_{u=1}^M \left(-R_u \log(\hat{R}_u) - (1 - R_u) \log(1 - \hat{R}_u) \right) + \frac{\lambda_2}{2} \left(\sum_{l=1}^L \|W_l^r\|_2^2 + \sum_{l=1}^L \|b_l^r\|_2^2 \right),$$

where $l \in \{1, \dots, L\}$, M denotes the total number of users in the user set. After the latent features of user auxiliary information and user rating information are extracted, we fused them together, i.e., $X_i = [H_s, H_u]$, and then input to the MLP network to further obtain user's latent feature representation P_i . Defined as Equation (5):

$$(5) \quad P_i = h_n(\dots h_2(h_1(W_1^p X_i + b_1^p) + b_2^p) \dots),$$

where $h(\cdot)$ represents a nonlinear function. In our method, we use the ReLU function. W_*^p represents the weight of each layer, and b_*^p represents the bias of each layer.

2.2.2. Item latent feature modelling.

– *The latent feature extraction of item rating information:* As shown in the upper right part of Fig. 1, assuming that the item rating SDAE module and the user rating SDAE module have the same number of neural network layers, they are both L_r layers, the encoder is the first $\frac{L_r}{2}$ layers and the decoder is the following $\frac{L_r}{2}$ layers. The encoder is responsible for mapping the corrupted input $\tilde{R}_v$ to latent representation $H_{L_r/2}$, the decoder is responsible for reconstructing the original item rating information R_v from the latent representation $H_{L_r/2}$.

We define Equation (6) to obtain the latent representation of any layer l and the output of L_r layer as follows:

$$(6) \quad \begin{aligned} H_1 &= g(W_1^{r'} \tilde{R}_v + b_1^{r'}) \\ H_l &= g(W_l^{r'} H_{r,l-1} + b_l^{r'}) \\ \hat{R}_v &= f(W_{L_r}^{r'} H_{r,L_r-1} + b_{L_r}^{r'}). \end{aligned}$$

We use $H_v = H_{L_r/2}$ to represent the latent feature of the item rating information, and $\hat{R}_v$ to represent the output of the SDAE module, i.e., the reconstruction

of original user rating information R_v . $\tilde{R}$ represents the corrupted item rating information.

The objective function is defined as Equation (7):

$$(7) \quad \mathcal{L}_{R_v} = \sum_{v=1}^N \left(-R_v \log(\hat{R}) - (1 - R_v) \log(1 - \hat{r}_v) \right) + \frac{\lambda_3}{2} \left(\sum_{l=1}^L \|W_l^{r'}\|_2^2 + \sum_{l=1}^L \|b_l^{r'}\|_2^2 \right),$$

where $l \in \{1, \dots, L\}$, N denotes the total number of items in the item set.

– *The latent feature extraction of item image information:* We apply another unsupervised deep learning model, called convolutional stacked denoising autoencoder (CSDAE), to extract latent features from item image information by using convolutional hidden layers to replace the fully connected layers in SDAE.

Given a clean image input V and a corrupted input $\tilde{V}$, the generation process of each hidden layer l is as follows:

If it is a convolutional layer, the mapping function of the convolutional layer is defined as:

$$(8) \quad H_l = g(W_l^v \otimes H_{l-1} + b_l^v),$$

where $\otimes$ represents convolution operation.

If it is a fully connected layer, the mapping function of the fully connected layer is defined as:

$$(9) \quad H_l = g(W_l^v H_{l-1} + b_l^v).$$

The output of L_v -th layer is as follows:

$$(10) \quad \hat{V} = f(W_{L_v}^v \otimes H_{L_v-1} + b_{L_v}^v).$$

We use $H_e = H_{L_v/2}$ to represent the latent feature of the item image information.

The objective function is defined as Equation (11):

$$(11) \quad \mathcal{L}_{V_v} = \sum_{j=1}^N \left(V_j - \hat{V}_j \right)^2,$$

where N denotes the total number of items in the item set.

– *The latent feature extraction of item description information:* We adopt SDAE to extract latent features in item description information. In this method, a bag of words method is used to represent the description text of the item.

We define Equation (12) to obtain the latent representation of any layer l and the output of L_r layer as follows:

$$(12) \quad \begin{aligned} H_l &= g \left(W_l^d H_{r,l-1} + b_l^d \right) \\ \hat{D} &= f \left(W_{L_d}^d H_{r,L_r-1} + b_{L_d}^d \right). \end{aligned}$$

We use $H_d = H_{L_r/2}$ to represent the latent feature of the item description information, and $\hat{D}$ to represent the output of the SDAE module, i.e., the reconstruction of original user description information D . $\tilde{D}$ represents the corrupted item description information.

The objective function is defined as Equation (13):

$$(13) \quad \mathcal{L}_D = \sum_{j=1}^N \left(-D_j \log(\hat{D}) - (1 - D_j) \log(1 - \hat{D}_j) \right) + \frac{\lambda_4}{2} \left(\sum_{l=1}^L \|W_l^d\|_2^2 + \sum_{l=1}^L \|b_l^d\|_2^2 \right),$$

where $l \in \{1, \dots, L\}$, N denotes the total number of items in the item set. After the latent features of item rating information, item image information and item description information are extracted, we fused them together, i.e., $Y_j = [H_v, H_e, H_d]$, and then input to the MLP network to further obtain item's latent feature representation Q_j which is defined as Equation (14):

$$(14) \quad Q_j = h_n (\dots h_2 (h_1 (W_1^q Y_j + b_1^q) + b_2^q) \dots).$$

2.3. Feature fusion process. After feature extraction, we get the latent feature vectors of users and items (i.e., P_i and Q_j). The feature fusion process is to combine the user's feature and the item's feature under the MLP framework.

As shown in the upper middle part of Fig. 1, we fused these latent features together, i.e., $Z_{i,j} = [P_i, Q_j]$, as the input of the MLP module. The output of the l -th hidden layer is defined as follows:

$$(15) \quad \begin{aligned} h_1 &= \sigma (W_1^m Z_{i,j} + b_1^m) \\ h_l &= \sigma (W_l^m h_{l-1} + b_l^m), \end{aligned}$$

where W_l^m represents the weight of each layer, and b_l^m represents the bias of each layer; $\sigma(\cdot)$ represents a nonlinear function. In this model, we use the ReLU function, that is, $\sigma(\cdot) = \max(0, x)$.

We use $MLP(Z_{i,j})$ to denote the output of deep neural network model with $Z_{i,j}$ as the input vector, defined as follows:

$$(16) \quad MLP(Z_{i,j}) = f(W_n (\dots h_1 (W_1^m Z_{i,j} + b_1^m) \dots) + b_n^m),$$

where n represents the number of hidden layers of the MLP module; $f(\cdot)$ represents the nonlinear function of the output layer. In this model, we use the Softmax function.

The loss function of the deep neural network *MLP* is as follows:

$$(17) \quad \mathcal{L}_M = \sum_{i=1}^M \sum_{j=1}^N (R_{i,j} - \hat{R}_{i,j})^2,$$

where $R_{i,j}$ represents the actual rating given by the user to the item, and $\hat{R}_{i,j}$ represents the predicted rating.

In order to fuse the latent features of multi-source heterogeneous data, we propose a novel regularization method by minimizing the distance between the latent feature vectors in the middle-hidden layers of the neural network within the module. The relationship is formally defined as follows:

$$(18) \quad F_{reg} = \frac{\alpha}{2} \sum_{(u,v) \in M \times N} (\|H_u - H_s\|_F^2 + \|H_v - H_e\|_F^2 + \|H_v - H_d\|_F^2 + \|H_e - H_d\|_F^2),$$

where α represents the weight, which is used to constrain the impact of the associated regularization. The larger the value of α , the more features are shared between multi-source heterogeneous information, because multi-source heterogeneous data will show user preferences from different perspectives in real life. Therefore, the optimal value of α should be determined from the experimental results. Here $\|\cdot\|_F$ is the Frobenius norm for regularization term, which is used to calculate the error between the reconstructed data and the real data.

Finally, the loss function of the entire model is:

$$(19) \quad \mathcal{L} = \mathcal{L}_M + \mathcal{L}_s + \mathcal{L}_{R_u} + \mathcal{L}_{R_v} + \mathcal{L}_{V_v} + \mathcal{L}_D + F_{reg}.$$

2.4. Rating prediction and top- k list recommendations. After the model is trained by minimizing the model objective function, we use Equation (20) to predict the rating of the unrated items.

$$(20) \quad \hat{R}_{i,j} = MLP(Z_{i,j}),$$

where $\hat{R}_{i,j}$ is the predicted rating of item j by user u_i . Then, the top- k items with the highest score are selected from the unrated items of user u_i to generate recommendation list for user u_i .

2.5. Optimization methodology. We use the SGD method to learn the parameters of the model, and the parameters in the model are updated as follows:

$$(21) \quad \phi' = \phi - \eta \frac{\partial \mathcal{L}}{\partial \phi},$$

where ϕ represents all the parameters in the model; η represents the learning rate.

3. Experiment. **3.1. Datasets.** We select two different data sets to validate the effectiveness of our framework. CiaoDVD dataset contains 7375 users and 99 749 items, with a total of 278 483 user ratings and 111 781 social connections. The density of ratings and relationships are 0.02% and 0.015%, respectively. Epinions dataset contains 42 289 users and 139 738 items, with a total of 664 824 user ratings and 355 813 social connections. The density of ratings and relationships are 0.035% and 0.041%, respectively.

3.2. Evaluation metrics. In order to better evaluate the recommendation quality of our framework, we adopt the commonly used protocols: $NDCG@k$ and $MAP@k$. The larger values of two metrics indicate better performance. For each user, we randomly select 100 samples as negative and combine with the user’s positive samples to calculate the evaluation metrics.

3.3. Experimental settings. To verify the performance of our framework, we consider the following compared baselines: PopRank, PMF [2], BPR [3], GBPR [4], CDAE [5] and CDL [6]. For fair comparison, we select the optimal parameter settings of each base line in the relevant references. We set the parameter settings of the user SDAE module and the item SDAE module to be the same, the number of network layers is 4, the dimension of the latent feature vector (the middlemost hidden layer) is 100, the masking noise $q = 0.2$, and the regularization coefficient is 0.01. The number of network layers of the item CSDAE module is 6, the regularization coefficient is 0.01, the Gaussian noise is 3, and the number and size of convolution kernels are set to 20 and (5.5), respectively. The parameter α is set to 0.2.

3.4. Results and analysis. **3.4.1. Comparisons of different algorithms.** In this experiment, we use $MAP@10$ and $NDCG@10$ to evaluate the recommendation performance and set the dimensions of latent feature vectors to {10, 20, 50, 100, and 150}. Experimental results are shown in Fig. 2 and Fig. 3.

From Fig. 2 and Fig. 3, we can see:

(1) PopRank method performs worst on the two datasets. The main reason is that the algorithm sorts the items according to their popularity, does not mine the deep-level preferences of users, and does not consider the side information of users or items.

(2) PMF model performs better than PopRank algorithm on the two datasets. E.g. comparison results on PopRank on CiaoDVD dataset, the $NDCG@10$ of PMF improved 11.558%. However, it is not as good as BPR and the GBPR, mainly because PMF model performs probability matrix factorization on the user item ratings, while the BPR algorithm and GBPR algorithm is a list-based recommendation algorithm, so top- k recommendation is better.

(3) The models based on autoencoders (CDAE, CDL, and our proposed) have better recommendation performance than the models that are not based on autoencoders (PMF, BPR, GBPR). The main reason is that the models based on autoencoders using neural network to learn deep nonlinear relationship of data

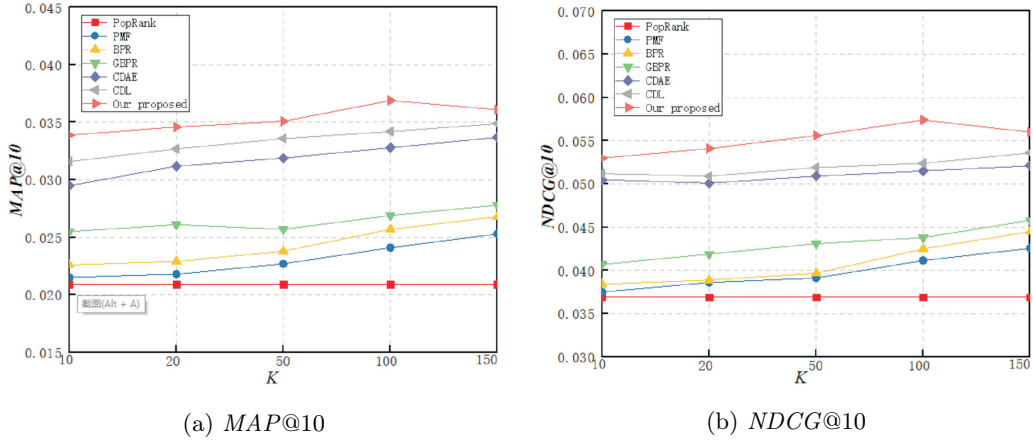


Fig. 2. Performance comparison of $MAP@10$ and $NDCG@10$ of different algorithm models on CiaoDVD dataset

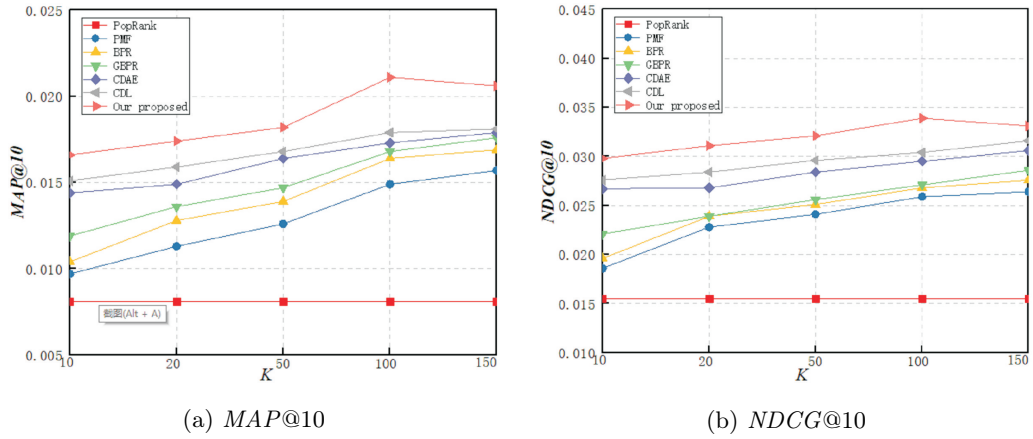


Fig. 3. Performance comparison of $MAP@10$ and $NDCG@10$ of different algorithm models on Epinions dataset

can better mine user's preference.

(4) On two different datasets, our framework outperforms other baselines and obtains the best results. E.g. comparison results on CDL on two datasets, the $NDCG@10$ of our proposed model improved 9.542% and 11.513%, respectively. This also verifies that our proposed method is effective in improving recommendation performance.

(5) As the dimension of the latent feature vector increases, the recommendation performance of our proposed method keeps improving, which indicates that it is easier to learn the user's latent preference by utilizing the larger dimension of the latent feature.

(6) Top- k recommendation performance of all models on the Epinions dataset

is poor. The main reason is that the user item rating data and auxiliary information data in the Epinions dataset are large, which will generate a certain noise interference and eventually lead to the conclusion that top- k recommendation is not ideal.

3.4.2. The effect of parameter α . In our proposed method, the parameter α has an important impact on the recommendation performance. It mainly controls the influence of the relevant regularization term. The larger the value of α , the better the correlation between auxiliary information and user rating information. We conducted a set of experiments on the CiaoDVD and Epinions datasets using the NDCG@10 metrics as an example, and the experimental results are shown in Fig. 4.

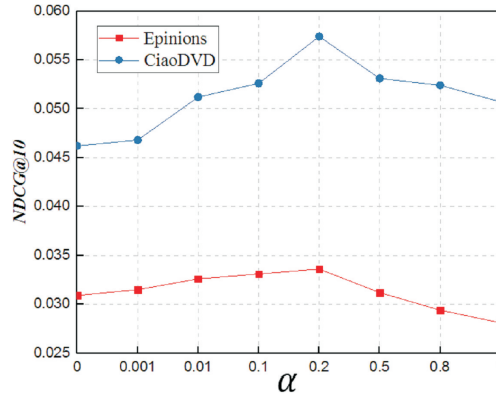


Fig. 4. The effect of parameter α on NDCG@10 on both datasets

From Fig. 4, we can see that the value of α has a great influence on the recommendation performance. With the change of α , $NDCG@10$ has changed a lot. When $\alpha < 0.2$, with the increase of α , $NDCG@10$ gradually increases until $\alpha = 0.2$, $NDCG@10$ reaches the highest value, when $\alpha > 0.2$, with the increase of α , the $NDCG@10$ value of the algorithm gradually decreases, which is due to the decrease in recommendation performance caused by overfitting. E.g., $NDCG@10$ are 0.0336 and 0.0574 on the CiaoDVD and Epinions datasets, respectively. Experimental results show that the fusion of auxiliary information has an important impact on the performance of the algorithm. Therefore, the optimal α value is taken as 0.2.

4. Conclusion and future work. From the perspective of auxiliary information fusion, we proposed a novel neural collaborative filtering recommendation based on side information fusion, which adopted deep learning to model the auxiliary data related to users and items, thereby solving the problem of sparse user historical interactions and the problem of cold start. Our proposed model contains five SDAE network structures, which are tightly connected by the relevant

regularization terms of the middlemost hidden layer. In particular, we proposed a tightly coupled regularization term that can mine latent preferences from relevant auxiliary data of users and items, while the model is structurally flexible and can be easily extended to other types of auxiliary information. Finally, we conducted different comparative validations on two different datasets, and comparison results demonstrated the excellent performance of our framework.

In future work, group recommendation is a trend in the development of recommender systems. How to analyze complex group relationships, group scenarios, group preferences, and build a more accurate and intelligent group recommendation model will be the future research direction.

REFERENCES

- [1] MU R. (2020) A novel recommendation model based on deep neural network, C. R. Acad. Bulg. Sci., **73**(5), 681–690.
- [2] MNIH A., R. SALAKHUTDINOV (2008) Probabilistic matrix factorization. In: Advances in neural information processing systems, 1257–1264.
- [3] RENDLE S., C. FREUDENTHALER, Z. GANTNER, L. SCHMIDT-THIEME (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal, Canada, 452–464.
- [4] PAN W., L. CHEN (2013) GBPR: group preference-based Bayesian personalized ranking for one-class collaborative filtering. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 2691–2697.
- [5] WU Y., C. DUBOIS (2016) Collaborative Denoising Autoencoders for Top-N Recommender Systems. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, New York, NY, USA, 153–162.
- [6] WANG H., N. WANG, D.-Y. YEUNG (2015) Collaborative deep learning for recommender systems. In: KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1235–1244.

*College of Computer and Information Engineering
Xinxiang University
Xinxiang, Henan
P. R. China, 453000
e-mail: muruihui@126.com*