# Integrating Global Features into Neural Collaborative Filtering

Langzhou He[1], Songxin Wang[2], Jiaxin Wang[3], Chao Gao[4](✉), and Li Tao[1](✉)

[1] College of Computer and Information Science, Southwest University, Chongqing 400715, China
tli@swu.edu.cn

[2] School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China
sxwang@mail.shufe.edu.cn

[3] State Grid Information & Telecommunication Group Co., Ltd., Beijing, China
wangjiaxin1@sgitg.sgcc.com.cn

[4] School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China
cgao@nwpu.edu.cn

**Abstract.** Recently, deep learning has been widely applied in the field of recommender systems and achieved great success, among which the most representative one is the Collaborative Filtering based Deep Neural Network. However, the input of such a model is usually a very sparse one-hot coding vector of users and items. This makes it difficult for the model to effectively capture the global features interaction between users and items. What is more, it also increases the training difficulty, making the model easily fall into a local optimum. Therefore, this paper proposes a two-stage Integrating Global Features into Neural Collaborative Filtering (GFNCF) model. To begin with, the AutoEncoder model with sparse constraint parameters is used to accurately extract the global features of users and items. Following that, the global features extracted in the previous step are integrated into the neural collaborative filtering framework as auxiliary information. It alleviates the sparse input problem and integrates more auxiliary features to improve the learning process of the model. Extensive experiments on several publicly available datasets demonstrate the effectiveness of the proposed GFNCF model.

**Keywords:** Deep learning · Recommender system · Collabrative filtering · AutoEncoder · Feature extraction

## 1 Introduction

Living in an "information society", recommender systems greatly solve the problem of information overload [5]. The task of the personalized recommender systems is to recommend users potential interest items according to their preferences, and the system has been widely applied in many fields such as news


G. Memmi et al. (Eds.): KSEM 2022, LNAI 13369, pp. 325–336, 2022.
https://doi.org/10.1007/978-3-031-10986-7_26

[3,6] and music [2,16]. Rating prediction is a main task in the personalized recommender systems. The key to the prediction is to model the characteristics of users and items according to the rating interaction between them, and realize the rating prediction of items (such as 1–5 stars) [19]. For recommender services, predicting users' preference for pushed items through rating predictions can improve users satisfaction effectively, and provide support for corporate decision-making and bring huge economic benefits to the company.

In recent years, Collaborative Filtering, acts as one of the effective means of rating prediction, has made big progress. Researchers applied all kinds of deep learning technology to the Collaborative Filtering and made it successfully [11]. Deep Neural Network has been widely used to extract High-level features in user-item interactive information [7,18]. The quality of these high-level interactive features decides the performance of the model directly most of the time. Although DNN has massive advantages in high-level features extraction and parameters learning, the input feature vector of this type of model is often extremely sparse due to the sparseness of the data in the recommender systems. In this case, the optimization result of the model is easy to fall into a local optimum, which affects the performance of the model.

Global features, known as the overall properties of the user image, are always difficult to be captured effectively since most Collaborative Filtering models based neural network ofter take the one-hot representation of users and items as input for feature learning. Studies have shown some global features of users and items can effectively improve the prediction accuracy of the model [13]. Therefore, some recent methods use neighborhood information or global information of users and items to replace one-hot representation to input into the neural network model for feature extraction [1,4,17]. Although this type of model uses neighborhood information or its own global information to enhance the interaction capabilities of the model to a certain extent, it also weakens the ability of user-project single interaction modeling.

Based on the considerations above, we find that alleviating the sparse input feature problem and making full use of user item interaction information can further improve the performance of the CF method. Therefore, this paper incorporates them into the proposed Integrating Global Features into Neural Collaborative Filtering (GFNCF) model. First, for the sparse input feature problem, we use an improved feature extraction method to extract the dense global feature vector of users and items accurately, and then we use the extracted global feature vector as an additional input to fuse neural collaborative filtering [8] in the framework, thus alleviating the training difficulty learning from sparse features of the model. Moreover, the different features in the fused model are crossed and combined through joint training, thereby effectively improving the accuracy of the final rating prediction. The main contributions of our work are as follows :

- This paper introduces the sparsity constraint parameters into the AutoEncoder-based collaborative filtering model to make the model more adaptable to the sparsity characteristics of the data in the recommender systems, and to accurately extract the global features of users and items.

– This paper integrates the global characteristics of users and items as auxiliary information in the neural collaborative filtering framework, which effectively expands the framework and further improves the performance of the method.
– This paper conducts extensive experiments on five real-world data sets to prove the effectiveness of the proposed method.

## 2 Integrated Global Features Neural Collaborative Filtering

The main process of our proposed method is shown in Fig. 1. It consists of two parts, namely the extraction of global features and the integration of global features. The extraction of global features part inputs the rating matrix of users and items into the AutoEncoder to extract the global features of each entity (user or item). These global features will be used for the global feature interaction modeling. The integration of global features part integrates extracted global features into the neural collaborative filtering model which makes the model capture the interaction between users and items more accurately.

In general, our proposed GFNCF model is a model-based method, which assumes that the rating data is generated by the model. To estimate the parameters in the model, existing methods generally adopt the method of optimizing the loss function in machine learning. In this paper, we use point-wise loss as the optimization loss function, that is, the parameters in the GFNCF model are estimated by minimizing the square loss between predicted value and true value.

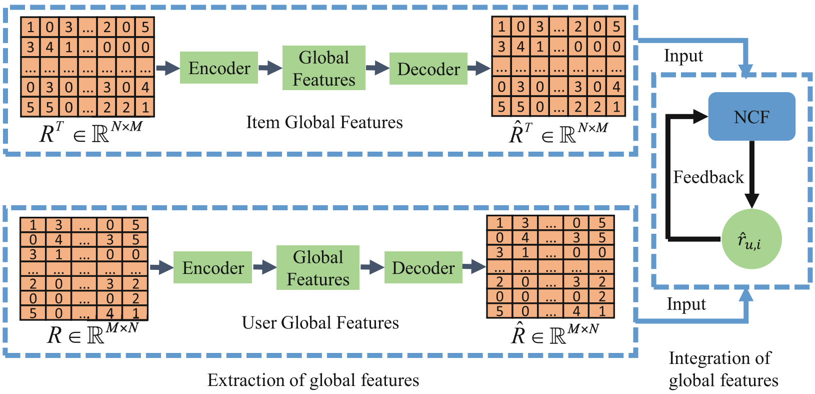


**Fig. 1.** The proposed method is divided into two parts.

### 2.1 Extraction of Global Features

Let $M$ and $N$ respectively represent the total number of users and items in the recommender systems, then the users set and items set can be expressed as $U = \{U_1, U_2, ..., U_M\}$ and $V = \{V_1, V_2, ..., V_N\}$. Construct a user-item rating matrix $R \in R^{M \times N}$ based on the explicit ratings feedback of users on items

$$R_{u,i} = \begin{cases} r_{u,i}, & if\ user\ u\ has\ rated\ item\ i \\ 0\ , & otherwise \end{cases} \tag{1}$$

where $r_{u,i}$ represents the rating of user $u$ on item $i$, $u \in \{1, 2, ..., M\}$ and $i \in \{1, 2, ..., N\}$. $R_{u,i}$ equals to 0 does not mean that user $u$ has rated item $i$ as 0. It may because user $u$ has not rated item $i$.

AutoEncoder is an unsupervised learning algorithm, mainly used for data dimensionality reduction or feature extraction [10]. In order to extract the global features of the item, the rating information of each item $i$ is expressed as:

$$r^{(i)} = (R_{1,i}, R_{2,i}, \ldots, R_{\mathrm{M},i}) \tag{2}$$

Taking the use of a deep autoencoder to extract the global features of item $i$ as an example, the feature extraction process of this model can be defined as:

$$g_f = g_l \left( W_l^T z_{l-1} + b_l \right) \tag{3}$$

where $g_l(\cdot)$, $W_l$, $b_l$ represent the activation function, parameter matrix, and bias vector of the $l$-th neural network, respectively, $g_f$ represents the extracted global features of item $i$.

The process of reconstructing rating information through the global features of item $i$ can be defined as:

$$\hat{r}^{(i)} = g_{2l-1} \left( W_{2l-1}^T z_{2l-2} + b_{2l-1} \right) \tag{4}$$

where $\hat{r}^{(i)}$ represents the rating information reconstructed from the global features of item $i$. Then we input the items rating vector $\left\{ r^{(i)} \right\}_{i=1}^{N}$ set into deep autoencoder model to obtain the reconstructed rating vector set. In order to make the reconstructed rating vector as close as possible to the original rating vector in observed rating part, the following loss function is used as the optimization target of the autoencoder model [7]:

$$\min_{\theta} \sum_{i=1}^{N} \left\| r^{(i)} - h(r^{(i)}; \theta) \right\|_O^2 + \lambda \|\theta\|_F^2 \tag{5}$$

where $h(r^{(i)}; \theta)$ represents the rating information of item $i$ reconstructed by the autoencoder, $\theta$ represents the parameters of the deep autoencoder model, and the $\lambda$ parameter is used to control the complexity of the model and prevent over-fitting together. $|| \cdot ||_O^2$ means that only the errors of the ratings observed in the training set are considered.

It is widely acknowledged that the overall item rating quantity information in the recommender system presents a long tail distribution [15]. For items with sparse ratings, the integration of global features will provide effective additional features for the model to fit these items and improve the learning process of the model. Therefore, we introduce a sparsity constraint parameter into the AutoEncoder model for extracting global features, which can help the model fit sparse items as much as possible, and provide effective auxiliary features for sparsely rated items. We introduce the sparsity constraint parameter to obtain the following loss function:

$$Loss = \sum_{i=1}^{N} \frac{\alpha}{\sum_{j=1}^{M} I\left(r_j^{(i)} \neq 0\right)} ||r^{(i)} - \hat{r}^{(i)}||_o^2 + \lambda||\theta||_F^2 \tag{6}$$

where $I(X)$ is the indicator function. If $x$ is true, then $I(x)$ is 1, otherwise $I(x)$ is 0. It can be noticed that when the rating information of an item is sparse, the model prediction is not accurate. The greater the error, the greater the penalty for the model. $\alpha$ is a constraint factor, and the value is generally greater than 1. By adjusting the value of $\alpha$ , we can control the degree to which the model tends to fit sparse data well. $\lambda$ is a regularisation parameter used to control model complexity and prevent overfitting of the model.

## 2.2 Integration of Global Features

After extracting the global features of users and items, the global features are transformed by Multi-Layer Perceptron ($MLP$) to obtain the final features that are concatenated into the neural collaborative filtering model for joint training. The additional input information not only alleviates the problem of sparse input features but also increases the ability of the neural collaborative filtering model to capture global feature interactions. The overall GFNCF model is shown in Fig. 2, which shows deep matrix decomposition model can be used to learn a more accurate representation for both user and item using a deep neural network. Therefore, similar to the network architecture in DeepCF, this paper uses a deep matrix decomposition architecture for user and item representation learning in GFNCF.

Figure 2 plots the three parts of the overall GFNCF model, namely the deep matrix decomposition representation learning part, the MLP matching function learning part, and the global feature integration part.

**Representation Learning.** In order to model a single user-item interaction feature, we input users and items into deep matrix factorization through one-hot encoding for representation learning. For the sake of generality, assume that the input user is $u$ and the item is $i$, which are encoded as $x_u$ and $x_i$ by one-hot later. Let the latent vector of user $u$ be denoted as $p_u$, and the latent vector of item $i$ as $q_i$. Since the neural network is used in the deep matrix factorization to replace the linear embedding operation in the traditional matrix factorization, the low-dimensional representation learning process of the user $u$ is:

$$p_u = a_K = g\left(W_K^T a_{K-1} + b_K\right) \tag{7}$$

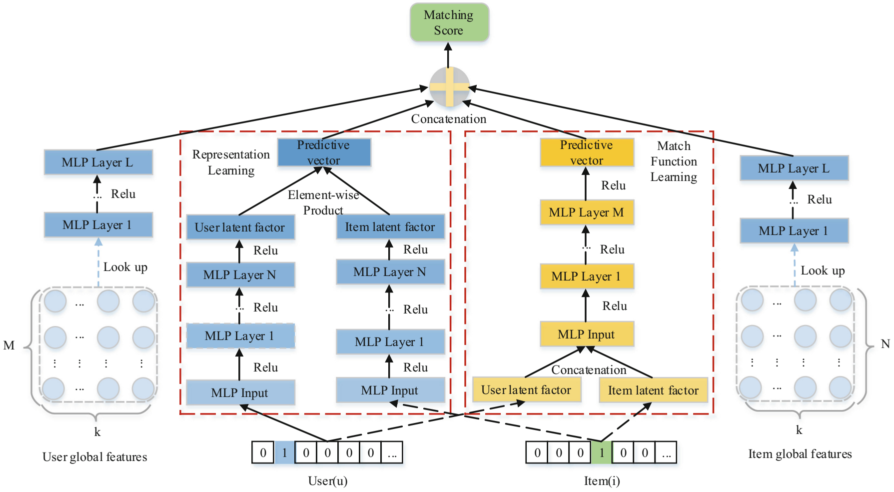


**Fig. 2.** The architecture of our proposed model.

where $W_l$, $b_l$ and $a_l$ represent the weight matrix, bias vector and output value of the $l$-th layer perceptron, respectively. $g(\cdot)$ is the activation function, this paper uses the $relu$ activation function.

In the same way, the low-dimensional representation learning process of item $i$ is:

$$q_i = a_K = g\left(W_K^T a_{K-1} + b_K\right) \tag{8}$$

We use the element-wise product of the low-dimensional representation vectors of users and items to calculate the feature vectors representing the learning part:

$$\phi_{MF} = p_u \odot q_i \tag{9}$$

where $\odot$ represents element-wise product.

**Matching Function Learning.** CF methods based on matching function learning usually use a linear embedding layer to learn low-dimensional latent representations of users and items, and then use dense low-dimensional latent representations to learn matching functions between users and items. We utilize MLP to learn the matching function in the user-item feature interaction. We assume $P$ and $Q$ respectively represent linear embedding layer parameter matrices that map $x_u$ and $x_i$ to low-dimensional dense representations. Therefore, the one-hot encoding of users and items in the embedding layer is converted into low-dimensional dense representation. The part can be described as:

$$p_u = P^T x_u \tag{10}$$

$$q_i = Q^T x_i \tag{11}$$

In order to fully cross the features and enhance the model's ability to model user-item interaction characteristics, in this paper, the obtained low-dimensional representation vectors of users and items are connected together and then sent to MLP for matching function learning.

$$\phi_{MLP} = a_M = g\left(W_H^T a_{H-1} + b_H\right) \tag{12}$$

**Global Features Learning.** Let $p_{gf}^u$ and $q_{gf}^i$ respectively represent the global feature vector of user $u$ and item $i$. In order to make the global features have better expressive ability in the GFNCF model, the GFNCF model sends the extracted global features into the MLP for feature transformation to transform them into a global feature representation that is more suitable for the model. The user's global feature embedding part of the model is as follows:

$$p_u^{GF} = a_E = g\left(W_L^T a_{L-1} + b_L\right) \tag{13}$$

The embedded part of the project's global features is as follows:

$$q_i^{GF} = a_L = g\left(W_L^T a_{L-1} + b_L\right) \tag{14}$$

Finally, the global characteristics of users and items are connected into a vector representation:

$$\phi_{GF} = \begin{bmatrix} p_u^{GF} \\ q_i^{GF} \end{bmatrix} \tag{15}$$

**Fusion of Feature Vectors.** The above three feature vectors express different forms of feature representation. In order to obtain a more accurate joint representation of user items, we need a strategy to fuse them so that they can enhance as well as interacting with each other, thereby enhancing the expressive ability of the model. The most common fusion strategy is to connect the obtained multiple feature vector representations to obtain a joint representation, and then input them into a fully connected layer for joint training. The fully connected layer can assign different weights to each feature contained in the joint representation. This adaptive combination of weight features makes the ultimate calculated matching scores of users and items more accurate, so the final output of the model in this paper is:

$$\hat{r}_{u,i} = W_{\text{out}}^T \begin{bmatrix} \phi_{MF} \\ \phi_{MLP} \\ \phi_{GF} \end{bmatrix} + b \tag{16}$$

where $W_{\text{out}}$ represents the weight parameter of the output layer of the model.

During the training process of the model, this paper uses the mean square error of the predicted rating $\hat{\boldsymbol{r}}_{u,i}$ and the true rating $r_{u,i}$ as the loss function to calculate the back propagation as well as optimizing the network parameters.

# 3 Experiments

## 3.1 Experimental Setup

**Datasets.** We evaluate our proposed models on five public datasets, i.e., movieLens (ml-la, ml-1m, ml-10m), filmtrust and ciaodvd in Table 1. To verify the effectiveness of the proposed model, this paper selects five publicly available datasets in the real world.

**Table 1.** The statistics of datasets. #User represents the number of users, #Items represents the number of items, #Ratings represents the number of ratings, #Sparsity represents the sparsity of datasets.

| Dataset | #Users | #Items | #Ratings | #Sparsity |
|---|---|---|---|---|
| ml-la | 610 | 9724 | 100836 | 98.30% |
| ml-1m | 6040 | 3706 | 1000209 | 95.53% |
| ml-10m | 69878 | 10677 | 10000054 | 99.69% |
| ciaodvd | 17615 | 16121 | 72665 | 99.97% |
| filmtrust | 1508 | 2071 | 35497 | 98.86% |

The statistical indicators of each dataset are given in Table 1. It is easy to see from the statistical indicators that there are differences in the scale of each dataset, therefore we can verify the universality of the model. In this paper, each dataset is divided randomly at a ratio of 4:1 into training set and test set for the following experiments.

**Evaluation Metrics and Baselines.** We compared the performance of our proposed GFNCF model with following six algorithms, namely, UserCF [9], PMF [14], DeepCF [8], NeuMF [4], DMF [18], BPAM [17].

**Table 2.** Comparison results on the public datasets evaluated by RMSE and MAE, where the best result is in bold.

| Dataset | Measure | UserCF | PMF | DeepCF | NeuMF | DMF | BPAM | GFNCF |
|---|---|---|---|---|---|---|---|---|
| ml-la | RMSE | 1.1072 | 0.9976 | 0.8727 | 0.8511 | 0.9004 | 0.8942 | **0.8409** |
| | MAE | 0.8329 | 0.7746 | 0.6791 | 0.6602 | 0.6974 | 0.6891 | **0.6527** |
| ml-1m | RMSE | 1.1777 | 0.9319 | 0.8837 | 0.8743 | 0.8816 | 0.9583 | **0.8647** |
| | MAE | 0.8687 | 0.7345 | 0.6885 | 0.6864 | 0.6875 | 0.7479 | **0.6693** |
| ml-10m | RMSE | 1.3105 | 0.9162 | NA | 0.8577 | 0.8684 | 0.9237 | **0.8327** |
| | MAE | 0.9218 | 0.7207 | NA | 0.6576 | 0.6817 | 0.7259 | **0.6431** |
| ciaodvd | RMSE | 1.3084 | 1.1947 | 1.0225 | 0.9507 | 1.0489 | 1.0241 | **0.9307** |
| | MAE | 0.9158 | 0.8663 | 0.8374 | 0.7393 | 0.8104 | 0.7932 | **0.7255** |
| filmtrust | RMSE | 1.0907 | 0.9724 | 0.8371 | 0.8002 | 0.87009 | 0.8212 | **0.7845** |
| | MAE | 0.8164 | 0.7453 | 0.6487 | 0.6284 | 0.6887 | 0.6457 | **0.6074** |

**Parameter Settings.** For the sake of fairness, we use the parameter settings from the original model in the neural collaborative filtering part of the GFNCF model whenever possible to assess the effectiveness of the proposed method. Therefore, this paper utilizes Gaussian distribution (mean value is 0, standard deviation is 0.01) to randomly initialize the model parameters, and uses mini-batch gradient descent and Adam [12] algorithm to optimize the model. Set the learning rate to 0.001 and the batch size to 1024 (set the batch size to 4096 on the ml-10m large dataset to speed up training). In addition, this paper sets the feature dimension size of users and items extracted from the neural collaborative filtering model to 32. Silimar to latent factor model, we defaults that the global features dimension of users and items is same. We set up different number of layers in neural networks to transform global features and 2–3 layers in the neural network can achieve great results.

### 3.2 Overall Comparison

This article uses root mean square error (RMSE) and mean absolute error (MAE) to evaluate the performance of the prediction results. The smaller the values of RMSE and MAE are, the better the performance is. The RMSE and MAE are defined as follows:

$$\begin{aligned} RMSE &= \sqrt{\frac{7}{N} \sum_{(u,i) \in R^+} (r_{u,i} - r_{u,i})^2} \\ MAE &= \frac{1}{N} \sum_{(u,i) \in R^+} |r_{u,i} - r_{u,i}| \end{aligned} \tag{17}$$

First, we can find that our proposed GFNCF method achieved an average improvement of 1.8% compared with the sub-optimal method (NeuMF). Note that NeuMF is a specific implementation under the neural collaborative filtering. These experimental results proved that integrates global features to neural collaborative filtering can effectively improve the performance of the model. Compared with DMF, the GFNCF model achieved a huge average improvement of 7.1% on all datasets. The main reason is that the GFNCF model adds auxiliary information of global features and matching function learning part to enhance the ability of model to capture the non-linear features of user-item interaction. In addition, compared with the DeepCF method, the GFNCF method models the user-item interaction information by combining the global features interactions with user-item interactions, while DeepCF only utilizes global information for modeling, ignoring the single interaction relationship between users and items. As a result, GFNCF achieved an average improvement of 3.7% compared with DeepCF. What's more, the results on MovieLens datasets (ml-1a, ml-1m and ml-10m) in Table 2 illustrates that as the data size and data sparsity increase, the improvement effect achieved by the GFNCF method gradually increases from 1.2% to about 3%. It shows that taking the global features as extra input alleviates the problem of sparsity input and effectively improve the performance of the model. In general, GFNCF alleviates the sparse input feature problem after fusing the global features and provides more feature information for model training, which improves the model performance to a certain extent.

### 3.3 Detailed Model Analysis

**Model Ablation Analysis.** In the design of GFNCF, we integrate the global features of users and items learned from the rating matrix into the existing framework to train the network jointly. In order to verify the effectiveness of users and items global features embedded in the model, we compared the model with the following three model variants: GFNCF-UF (without users global features in GFNCF), GFNCF-IF (without items global features GFNCF), NCF (GFNCF without users and items global features). As shown in Table 3, we have the following observation results: (1) The addition of items global features and users global features in the five datasets has improved the effect of the algorithm to a certain extent. (2) As the dataset changes, the optimizing degree on the model slightly differentiates. (3) When the global features of users and items are added at the same time, we have the best performance of the model. The reason for above results is that the more the global features of the user and the item are crossed and combined in the neural network, the better the learning process of the model, which leads to the best performance of the proposed method.

**Table 3.** The comparison results of different variants of GFNCF model evaluated by RMSE and MAE, and the best results are shown in bold.

| Dataset | Measure | NCF | GFNCF-UF | GFNCF-IF | GFNCF |
|---|---|---|---|---|---|
| ml-la | RMSE | 0.8521 | 0.8491 | 0.8445 | **0.8409** |
| | MAE | 0.6602 | 0.6506 | 0.6511 | **0.6527** |
| ml-1m | RMSE | 0.8743 | 0.8671 | 0.8705 | **0.8647** |
| | MAE | 0.6864 | 0.6729 | 0.6788 | **0.6693** |
| ml-10m | RMSE | 0.8577 | 0.8512 | 0.8453 | **0.8327** |
| | MAE | 0.6576 | 0.6537 | 0.6489 | **0.6431** |
| ciaodvd | RMSE | 0.9517 | 0.9438 | 0.9487 | **0.9307** |
| | MAE | 0.7403 | 0.7332 | 0.7368 | **0.7255** |
| filmtrust | RMSE | 0.8002 | 0.7973 | 0.7904 | **0.7845** |
| | MAE | 0.6284 | 0.6184 | 0.6105 | **0.6074** |

**Influence of Sparsity Constraint Parameters.** We utilize the loss function shown in Eq. 6 to extract the global features of users and items. The most important parameter in this loss function is sparsity constraint parameters. By setting different values of $\alpha$, the degree to which the model tends to fit sparse data can be controlled. We integrate the global features extracted from different values of $\alpha$ into the model for comparative experiments. As shown in Fig. 3, except for the filmtrust dataset, all other datasets have the best RMSE performance in the range of user $\alpha$ and item $\alpha$ between 4–6. On filmtrust dataset, the best performance can be obtained when user $\alpha$ and item $\alpha$ are within the range of 2–4. Comparing the experimental results on the ml-1m and ml-10m datasets, it can be seen that when the sparsity and the size of the dataset increases, the proper enhancement of the sparsity constraint parameters of the AutoEncoder can improve the model performance.

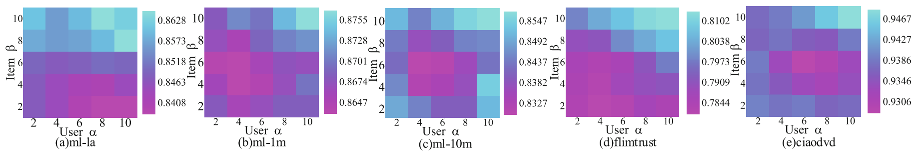


**Fig. 3.** RMSE index of the model after integrating global features extracted from different $\alpha$

## 4 Conclusion

This paper proposed an integrated global features neural collaborative filtering model which captured both user-item and user-item global feature interaction information and alleviated the sparseness problem of inputting feature vectors, avoiding the model training fall into a local optimum. Meanwhile, the paper utilized a kind of auxiliary information to develop another model with high performance. It is worth mentioning that the method used to extract auxiliary information should be as simple and effective as possible, so as to avoid adding too much additional load to the integrated model. Experimental results showed that the method in this paper has achieved certain advantages over traditional methods on datasets of different scales. Moreover, with the increase of data sparsity and data size, the advantages of this method become more apparent. In the future, we will use more auxiliary information, such as user portraits, comments, images and social information, as well as adding more data sources to explore better feature extraction methods for further research.

**Acknowledgement.** This work was supported by the National Key R&D Program of China (No. 2019YFB2102300), National Natural Science Foundation of China (No. 61976181, 11931015), Fundamental Research Funds for the Central Universities (No. D5000210738).

## References

1. Bai, T., Wen, J.R., Zhang, J., Zhao, W.X.: A neural collaborative filtering model with interaction-based neighborhood. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1979–1982 (2017)

2. Chen, H.C., Chen, A.L.: A music recommendation system based on music data grouping and user interests. In: Proceedings of the 10th International Conference on Information and Knowledge Management, pp. 231–238 (2001)
3. Chen, X., Li, S., Li, H., Jiang, S., Qi, Y., Song, L.: Generative adversarial user model for reinforcement learning based recommendation system. In: International Conference on Machine Learning, pp. 1052–1061 (2019)
4. Deng, Z.H., Huang, L., Wang, C.D., Lai, J.H., Philip, S.Y.: Deepcf: a unified framework of representation learning and matching function learning in recommender system. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 61–68, no. 01 (2019)
5. Edmunds, A., Morris, A.: The problem of information overload in business organisations: a review of the literature. Int. J. Inf. Manag. **20**(1), 17–28 (2000)
6. Gao, C., Liu, J.: Network-based modeling for characterizing human collective behaviors during extreme events. IEEE Trans. Syst. Man Cybern. Syst. **47**(1), 171–183 (2016)
7. Gupta, U., et al.: The architectural implications of facebook's DNN-based personalized recommendation. In: IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 488–501 (2020)
8. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
9. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237 (1999)
10. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
11. Huang, L., Jiang, B.L.S., et al.: Survey on deep learning based recommender systems. Chin. J. Comp. **41**(7), 1619–1647 (2018)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
14. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2008)
15. Park, Y.J., Tuzhilin, A.: The long tail of recommender systems and how to leverage it. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 11–18 (2008)
16. Pereira, B.L., Ueda, A., Penha, G., Santos, R.L., Ziviani, N.: Online learning to rank for sequential music recommendation. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 237–245 (2019)
17. Xi, W.D., Huang, L., Wang, C.D., Zheng, Y.Y., Lai, J.: BPAM: Recommendation based on BP neural network with attention mechanism. In: International Joint Conference on Artificial Intelligence, pp. 3905–3911 (2019)
18. Xue, H.J., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: International Joint Conference on Artificial Intelligence, pp. 3203–3209 (2017)
19. Zhang, S., Yao, L., Sun, A., Wang, S., Long, G., Dong, M.: Neurec: on nonlinear transformation for personalized ranking. arXiv preprint arXiv:1805.03002 (2018)