



Deep Matrix Factorization for Learning Resources Recommendation

Tran Thanh Dien¹, Nguyen Thanh-Hai¹, and Nguyen Thai Nghe^(✉)

College of Information and Communication Technology, Can Tho University,
Can Tho, Vietnam

{[thanhdien](mailto:thanhdien@cit.ctu.edu.vn),[nthai.cit](mailto:nthai.cit@ctu.edu.vn)}@ctu.edu.vn

ntnghe@cit.ctu.edu.vn

Abstract. Learning should last all through people's lives. With traditional learning, learners can meet face-to-face with their teachers or tutors. However, in some circumstances, learners cannot interact with their teachers. Learning resources (e.g., books, journals, slides, etc.) would be helpful for learners to get knowledge. With a large number of learning resources, how to select appropriate learning resources to learn is very important. In this work, a deep matrix factorization model extended from the standard matrix factorization is proposed for learning resources recommendation. We validate the proposed model on five published learning resources datasets and compare it with other well-known methods in recommender systems. The experimental results show that the proposed deep matrix factorization model works well, especially it can be a good choice for large-scale datasets.

Keywords: Learning resources recommendation · Deep learning · Knowledge search · Book recommendation · Matrix factorization · Deep matrix factorization

1 Introduction

Learning resources and materials are an important part of education. Learners and educators can be benefited from using them. Firstly, learning resources provide various digital learning environments such as open textbooks, open visual materials, open courses, and self-assessment tools. Then, numerous high-quality learning resources are possible to be updated and edited frequently. Another advantage of learning resources is that self-study, testing, or group learning are supported. With the development of information technology, it tends to move from traditional to online reading. Nowadays, information can be searched or exploited from e-libraries or on the Internet. Students and educators tend to search for their preferred learning materials on online resource systems. Universities have also provided open educational resources. They may consider an investment in open resources a sustainable human development. Open learning resources increase access to high-quality education and lower the cost of education worldwide. People around the world can share, contribute and access

knowledge. These resources refer to all aspects of social life, helping readers to exploit aspects and information about the problem they need to research. However, various educational resource and material systems may cause difficulties for users to select appropriate learning resources. With the explosion of e-library websites, volumes of data have been increasing. Today, most resources and materials system websites allow their users to write comments or/and reviews on the materials to express their feelings in a more precise and more detailed way about the materials they had read. This information enables us to conduct studies based on users' ratings for providing valuable suggestions.

In this study, a method based on deep learning and the state-of-the-art matrix factorization models is proposed to provide recommendations on learning resources. The work includes multi-fold contributions as follows:

- Propose a deep matrix factorization model with detailed architecture and parameters for learning resources recommendation.
- Compare the proposed model with other well-known methods in recommender systems to validate the results.
- Test on five learning resource datasets and compare the results of original data with pre-filtered data.
- Show that the deep matrix factorization model, a dimensional reduction approach, could be a good choice for large-scale datasets.

Section 2 presents some state-of-the-arts related to book recommendation systems. In Sect. 3, we present our proposed methods for this work. The experimental results of our proposed methods obtained are discussed concerning different parameters in Sect. 4. This section is followed by Sect. 5 containing the paper's conclusion.

2 Related Work

Numerous studies have been proposed to design automatic book recommendation systems using various methods, including classic machine learning, deep learning techniques, and hybrid methods on vast datasets that can range from a few features or even hundreds of attributes.

[1] stated that existing recommendation systems skipped the characteristics of readers' personalized information. Hence, their work proposed a library recommendation system based on a restricted Boltzmann machine and collaborative filtering algorithm to improve the performance to provide a good application effect. The work of [2] provided a system for exploring the data concerning to reading, including three categories.

There are numerous proposed methods and models, including user/item collaborative filtering, filtering based on content, association rule mining, hybrid recommender system, as well as recommendations. Though such techniques have exposed interesting characteristics, those can be inefficient in generating appropriate recommendations in some particular cases. Therefore, numerous hybrid

approaches have been investigated to integrate various methods to perform better recommendations [3]. A hybrid-based method presented in [4] has integrated to use both attribute and personality of users for book recommendation systems. Another hybrid-based method was introduced by [5] to consider customer's demographic information such as sex, age, geographical location, and book information such as title and ISBN for the rating of book prediction with values ranging from 1 to 5 classified with Multilayer Perceptrons models. A model learned from learner profile, and the learning content was proposed by [6] using two ontologies.

[7] considered that contextual information including location and emotion could enhance greatly product or service recommendations to be included in the recommender system. The authors determined several user characteristics and product features. In some studies, students' scores or courses are used as learning resources for the recommendation using deep learning techniques. [8] leveraged some techniques such as Quantile Transformation, MinMax Scaler to perform the prediction tasks. Another research of [9] proposed to use MultiLayer Perceptron and pre-processing methods on four million mark records of a university to give appropriate recommendations on course selection.

Based on the previous works as mentioned above, in this study, we propose an approach based on deep learning and the state-of-the-art matrix factorization models to provide good recommendations on learning resources (e.g., books, journals, etc.). We also compare the proposed model with other well-known recommender systems to prove state-of-the-arts of the proposed model.

3 Proposed Method

3.1 Problem Formulation

Let denote u as the user/learner, i as the learning resource (e.g., book, journal, paper, etc.), and r is the feedback from user u on learning resource i (rating). In general, the learning resources recommendation can be mapped to rating prediction problem in recommender systems as follows:

Learner, reader, or student \mapsto User
 Learning resource (book, journal, or paper, etc.) \mapsto Item
 Feedback (rating, number of views or clicks, etc.) \mapsto Rating

- **Prediction phase:** Given a dataset \mathcal{D} with available (u, i, r) We want to build the model to predict the rating (score) of the learning resources which have not been seen/read by the user (the empty values in this matrix).
- **Recommendation phase:** After having the prediction results, we sort the rating scores by ascending and selecting top N learning resources with the highest scores for recommendation (N could be 3, 5, or other value depending on the system interface).

3.2 Deep Matrix Factorization for Learning Resource Recommendation

Based on previous works [10, 11], in this study, we proposed the prediction model using Deep Matrix Factorization (DMF) that is described in detail in Fig. 1.

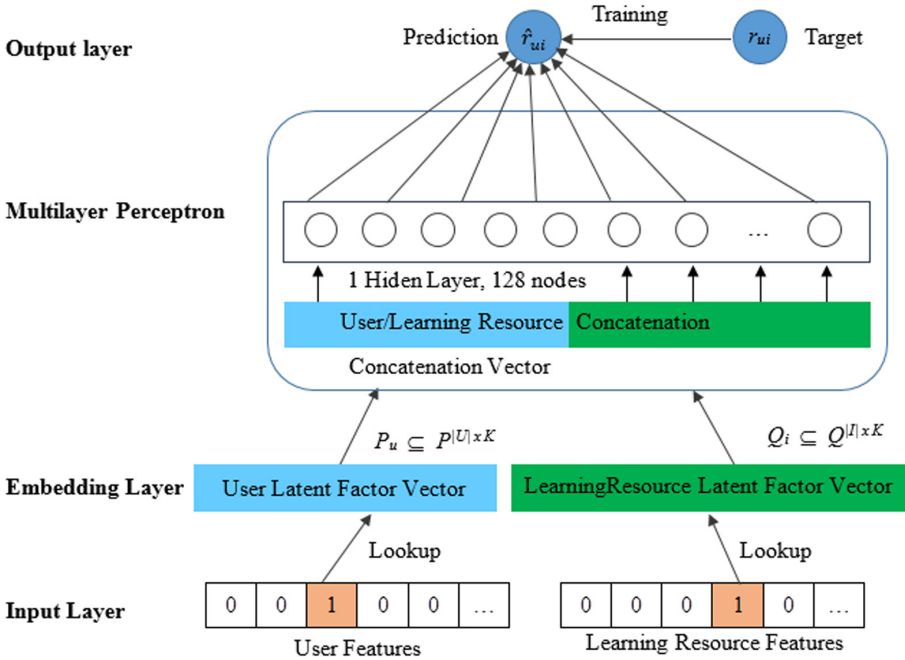


Fig. 1. Framework of Deep Matrix Factorization

The proposed model has four layers. An input layer represents the current user/learning resource; an embedding layer for embedding user and learning resource features. These embedding features are concatenated as the input for the Multilayer Perceptron (MLP) layer. Finally, an output layer for the prediction score. The MLP has 128 nodes (neurons). However, we can set up different numbers of hidden layers (e.g., adding more layers) and the different number of neurons dependent on different datasets/domains. In this study, the number of nodes is selected using hyper-parameter search, which will be carefully presented in the experimental results section (Sect. 4.4). The network uses Adam optimizer function, using a batch size of 256 while the learning rate is 0.001. Moreover, to explore the difference between the proposed DMF model and the standard Matrix Factorization (MF), we present the MF in Fig. 2. Instead of using an MLP as the DMF, the standard MF only does a DOT product between two users and learning resource feature vectors.

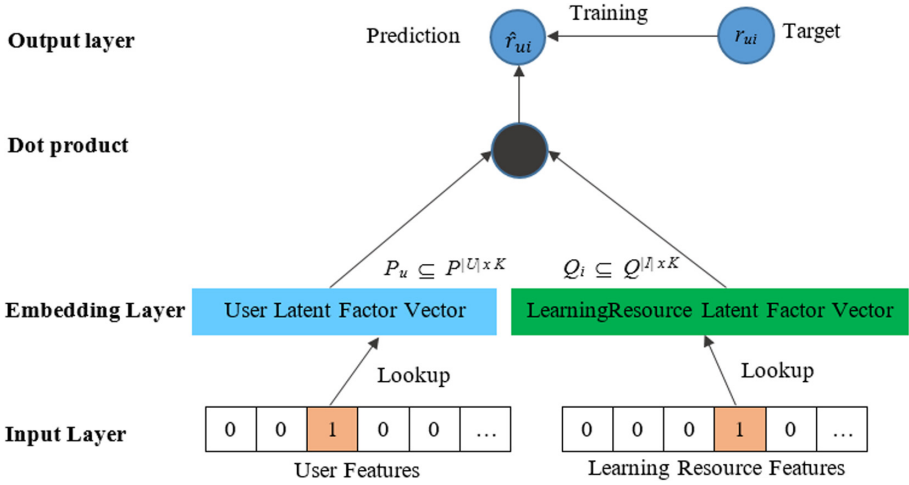


Fig. 2. Framework of Matrix Factorization

Table 1. Descriptions of five considered full datasets

No.	Dataset	#user	#item	#ratings
1	Ratings	53,424	10,000	981,756
2	LibraryThings	70,618	385,251	1,387,125
3	BX-Book-ratings	105,283	340,556	1,149,780
4	Related-Article Recommendation	2,663,825	7,224,279	48,879,167
5	Ratings-Books	8,026,324	2,330,066	22,507,155

Table 2. Descriptions of five considered datasets for at least 5 ratings per User/Learning Resource

No.	Dataset	#user	#item	#ratings
1	Ratings	32,492	10,000	916,880
2	LibraryThings	25,930	41,900	802,957
3	BX-Book-ratings	19,109	34,751	573,305
4	Related-Article Recommendation	2,368,923	2,225,631	37,145,643
5	Ratings_Books	622,558	596,401	9,389,719

4 Experimental Results

4.1 Data Description

1. **Ratings** dataset¹ contains all users' ratings of the books (a total of 980,000 ratings, for 10,000 books, from 53,424 users). It introduces and demonstrates collaborative filtering; allows us to take a deeper look at data-driven book recommendations.
2. **BX-Book-ratings** dataset² contains a collection of book ratings including 105,283 users providing 1,149,780 ratings (explicit and implicit) of about 340,556 books.
3. **Ratings-Books** dataset³ contains product ratings from Amazon, updated in 2018, including user, item, rating, and timestamp tuples.
4. **Related-Article Recommendation** dataset⁴ is based on data from a recommender system in the digital library and reference management software domain. The datasets are from the books. It contains 2,663,825 users, 7,224,279 books and 48,879,167 ratings.
5. **LibraryThings** dataset⁵ contains ratings and social relationships of users. It contains 70,618 users, offering 1,387,125 ratings of 385,251 books. In the dataset, the combination of social relations and rating similarity achieved the best results.

The number of users, learning resources, and ratings of these datasets are described in Table 1.

These datasets are very sparse, which means that the users or learning resources may have a few ratings. This problem is a challenge for every machine learning method. For comparison purposes, we present other versions of these datasets by keeping those users/learning resources with at least five ratings. The new version of these datasets is presented in Table 2.

4.2 Baselines for Comparison

In order to evaluate the effectiveness of the proposed DMF model, we compare it with other popular baselines for recommendation systems including User KNN, Item KNN, Co-Clustering, Global Average, User Average, Item Average, and Matrix Factorization.

Let denote u as the user/learner, i as the learning resource (e.g., book, journal, paper, etc.), and r is the feedback from user u on learning resource i (rating). These baselines can summarize as follows:

¹ <https://www.kaggle.com/philippsp/book-recommender-collaborative-filtering-shiny>.

² <https://www.kaggle.com/ruchi798/bookcrossing-dataset>.

³ <https://jmcauley.ucsd.edu/data/amazon/>.

⁴ <https://doi.org/10.7910/DVN/AT4MNE>.

⁵ <https://cseweb.ucsd.edu/~jmcauley/datasets.html>.

The Global Average method generates rating prediction (\hat{r}_{ui}) for user u on learning resource i by averaging the ratings in training dataset using Eq. 1 [12, 13].

$$\hat{r}_{ui} = \frac{\sum_{r \in \mathcal{D}^{train}} r}{|\mathcal{D}^{train}|} \quad (1)$$

The User Average method generates rating prediction (\hat{r}_{ui}) for user u on learning resource i by using Eq. 2 [12, 13].

$$\hat{r}_{ui} = \frac{\sum_{(u', i, r) \in \mathcal{D}^{train} | u' = u} r}{|\{(u', i, r) \in \mathcal{D}^{train} | u' = u\}|} \quad (2)$$

The Item Average is Learning Resource Average performing the predictions of the rating for the user u on learning resource i by using Eq. 3 [12, 13].

$$\hat{r}_{ui} = \frac{\sum_{(u, i', r) \in \mathcal{D}^{train} | i' = i} r}{|\{(u, i', r) \in \mathcal{D}^{train} | i' = i\}|} \quad (3)$$

The User KNN generates the rating prediction by using Eq. 4 [12, 13].

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{u' \in K_u} sim(u, u') \cdot (r_{u'i} - \bar{r}_{u'})}{\sum_{u' \in K_u} |sim(u, u')|} \quad (4)$$

where K_u is the k-nearest neighbors of user u and $sim(u, u')$ is the Cosine similarity between two users [12, 13], which is calculated by

$$sim_{cosine}(u, u') = \frac{\sum_{i \in I_{uu'}} r_{ui} \cdot r_{u'i}}{\sqrt{\sum_{i \in I_{uu'}} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_{uu'}} r_{u'i}^2}} \quad (5)$$

where $I_{uu'}$ is a set of learning resources which have rated by both user u and u' .

The Item KNN (Learning Resource KNN) generates the rating prediction by using Eq. 6 [12, 13].

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{i' \in K_i} sim(i, i') \cdot (r_{ui'} - \bar{r}_{i'})}{\sum_{i' \in K_i} |sim(i, i')|} \quad (6)$$

where K_i is the k-nearest neighbors of learning resource i and $sim(i, i')$ is the Cosine similarity between two learning resources, which is calculated by

$$sim_{cosine}(i, i') = \frac{\sum_{u \in U_{ii'}} r_{ui} r_{u'i'}}{\sqrt{\sum_{u \in U_{ii'}} r_{ui}^2} \cdot \sqrt{\sum_{u \in U_{ii'}} r_{u'i'}^2}} \quad (7)$$

where $U_{ii'}$ is a set of users who studied/rated both learning resource i and i'

The Co-Clustering method generates the rating prediction by using Eq. 8 [14].

$$\hat{r}_{ui} = \bar{C}_{ui} + (\mu_u - \bar{C}_u) + (\mu_i - \bar{C}_i) \quad (8)$$

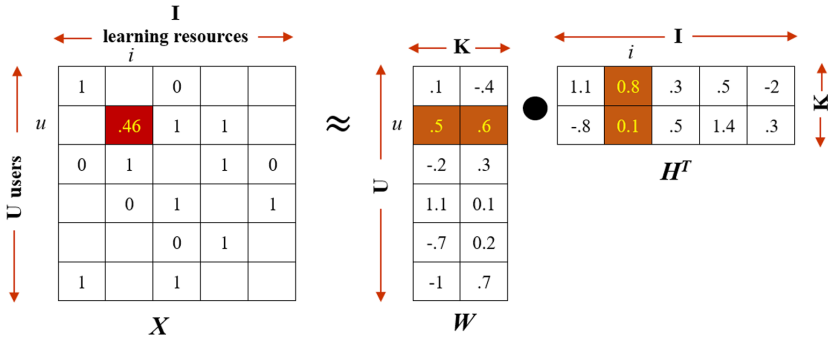


Fig. 3. Diagram of matrix factorization decomposition

where \bar{C}_{ui} is the average rating of co-cluster C_{ui} , \bar{C}_u is the average rating of u 's cluster, and \bar{C}_i is the average rating of i 's cluster. If the user is unknown, the prediction is $\hat{r}_{ui} = \mu_i$. If the learning resource is unknown, the prediction is $\hat{r}_{ui} = \mu_u$. If both the user and the learning resource are unknown, the prediction is $\hat{r}_{ui} = \mu$.

Matrix Factorization has been used widely in a vast of studies [12,13,15]. A matrix X is decomposed to two sub-matrices W and H as exhibited in Eq. 9:

$$X \approx WH^T \tag{9}$$

where $W \in \mathbb{R}^{|U| \times K}$ while $H \in \mathbb{R}^{|I| \times K}$ and K is latent factors, $K \ll |U|$, $K \ll |I|$. The Eq. 9 demonstrated as Fig. 3.

4.3 Evaluation Metrics

To compare the perform, the root mean squared error (RMSE) is used to evaluate the models. It calculated by Eq. 10 below:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{10}$$

where y_i is the true value, and \hat{y}_i is the predicted value.

4.4 Evaluation Results

The experimental results include an investigation on the impacts of the number of neurons (Fig. 4), the number of latent factors on the recommendation performance (Fig. 5), the performance of phases of training and testing during the learning (Fig. 6). Moreover, the results of various methods on the five considered datasets are exhibited in Figures of 7, 8, 9, 10, and 11.

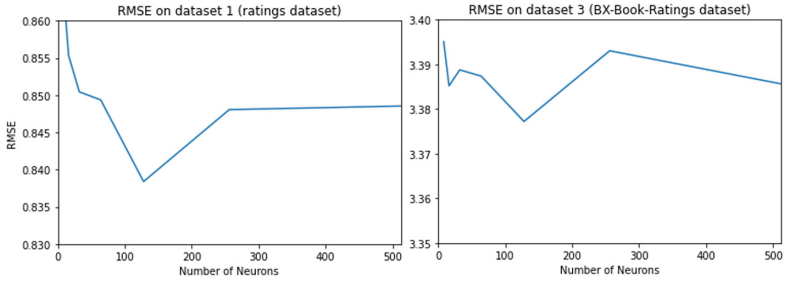


Fig. 4. The chart shows relationship between Number of Neurons for DMF models and RMSE performance

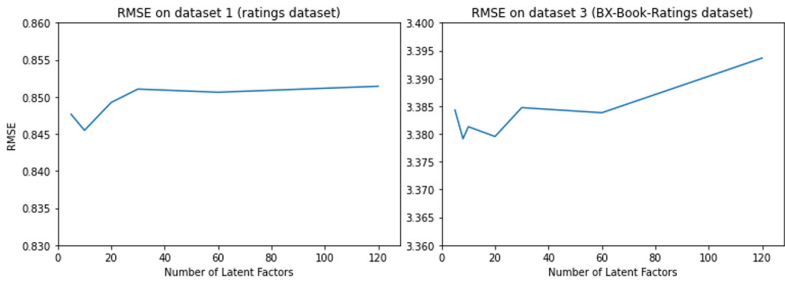


Fig. 5. The chart shows relationship between Number of Latent Factors (Features) for DMF models and RMSE performance.

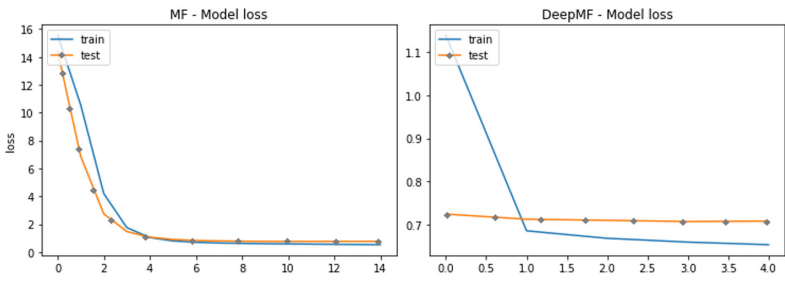


Fig. 6. Comparison of train loss and validation loss of the MF and DMF on Ratings dataset

In this study, the five published datasets are used for the experiments as follows.

The relationship between the error rate and the number of neurons for DMF models is exhibited in Fig. 4 while the effect of the Number of Latent Factors (Features) for DMF models on the error rate prediction is shown in Fig. 5. Two charts in Fig. 4 exhibit similar patterns in the performance. The number of neurons can increase the performance to a peak, then the loss goes up and tends

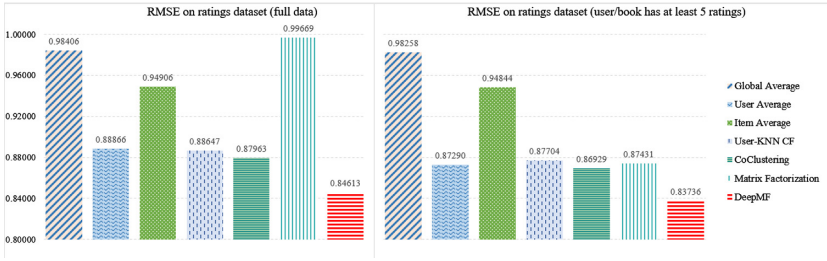


Fig. 7. RMSE comparison on dataset 1

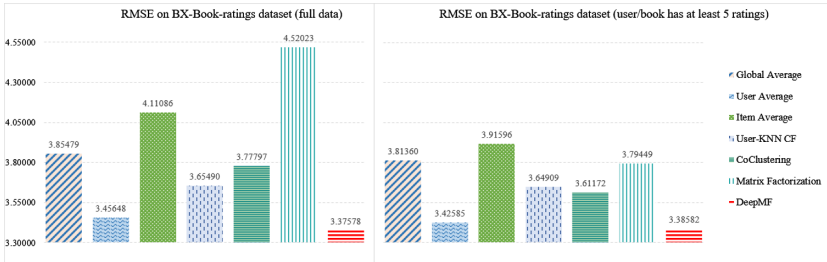


Fig. 8. RMSE comparison on dataset 2

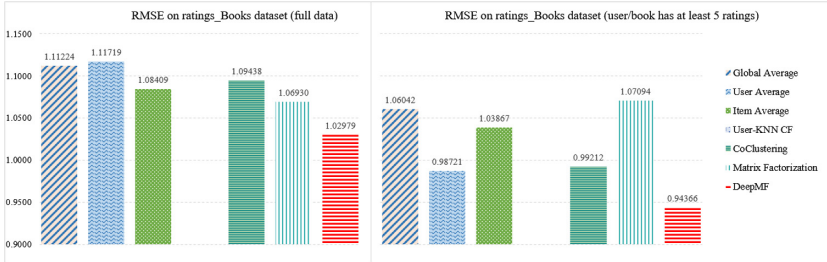


Fig. 9. RMSE comparison on dataset 3



Fig. 10. RMSE comparison on dataset 4

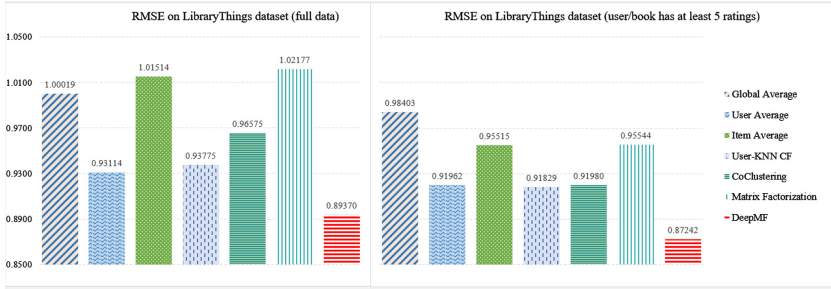


Fig. 11. RMSE comparison on dataset 5

to be saturated. On dataset 1, the lowest RMSE is achieved with around 100 neurons. A similar result is also obtained on dataset 3.

Figure 6 evaluates the overfitting of approaches during the learning. As shown in the figure, the training error and testing error of MF tends to converge after four epochs while DeepMF remains stable, the error rate only after two epochs. Moreover, MF suffers less overfitting than DeepMF when the learning progresses further.

Also, we compare DeepMF to a variety of methods including Item Average, User-KNN CF, Global Average, User Average, CoClustering, and Matrix Factorization in RMSE metric shown in Fig. 7 on dataset 1, Fig. 8 on dataset 2, and Fig. 9 on dataset 3, Fig. 10 on dataset 4, and Fig. 11 on dataset 5. For the datasets of 2 and 3, although we have deployed the experiments on the server with 320 GB in RAM, the learning of User-KNN CF cannot be done on two datasets of 3 and 4 due to the limitation of memory resources. In general, DeepMF outperforms most of the previous state-of-the-arts. Some better results can be obtained when we only consider the users/books with at least five ratings. Matrix Factorization seems to be the worst among the considered methods with the highest error rates on full datasets 1 and 5 and on dataset 3, including books/users with at least five ratings. The performance of User Average is almost near to DeepMF, but it only gets a better result on full dataset four and is defeated on other cases. From the prediction results, online learning resource systems can sort the resources/materials based on the rating scores corresponding to each user and then provide appropriate resources/materials suggestions to him or her as mentioned in the work of [9]. If a user logs into the system, it can extract the top five or top ten resources and materials with the highest predicted rating for that user. In the case the user does not log in, the systems can provide the top five or ten resources and materials which obtain the highest average predicted rating for all users.

5 Conclusion

In this work, we have proposed a DMF model extended from the standard matrix factorization for learning resources recommendations. We validate the proposed

model on five public published learning resources datasets and also compare it with other well-known baselines of recommender systems. The experiments show that the proposed model can obtain well results on various datasets. Primarily, it can be a good choice for large-scale datasets. However, in this work, some methods can not be done due to requiring a large amount of memory for running, e.g., User-KNN CF as revealed in the experiments. In future work, adding more information to the deep learning model (e.g., time, sequence, multi-relational effects) could be exciting for improving the model's performance.

Availability of Data, Codes, and Materials

The experimental scripts and a data sample of this study are published at **learning-resources-rs**: <https://github.com/thnguyencit/learning-resources-rs>.

References

1. Wang, X., Huang, H.: Research on library personalized recommendation system based on restricted Boltzmann machine. In: 2020 5th International Conference on Education and Social Development (ICESD 2020), pp. 293–297 (2020). <https://doi.org/10.12783/dtssehs/icesd2020/34428>
2. Xu, H., Ren, S., Yan, J., Li, Z.: Research and design of recommendation in book circle system. *J. Phys. Conf. Ser.* **1629**(1), 012063 (2020). <https://doi.org/10.1088/1742-6596/1629/1/012063>
3. Passi, R., Jain, S., Singh, P.K.: Hybrid approach for recommendation system. *Adv. Intell. Syst. Comput.* **828**, 117–128 (2019). https://doi.org/10.1007/978-981-13-1610-4_12
4. Hariadi, A.I., Nurjanah, D.: Hybrid attribute and personality based recommender system for book recommendation. In: 2017 International Conference on Data and Software Engineering (ICoDSE), pp. 1–5. IEEE (2017). <https://doi.org/10.1109/ICoDSE.2017.8285874>
5. Nirwan, H., Verma, O.P., Kanojia, A.: Personalized hybrid book recommender system using neural network. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1281–1288 (2016)
6. Aissaoui, O.E., Oughdir, L.: A learning style-based Ontology Matching to enhance learning resources recommendation. In: 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), pp. 1–7 (2020). <https://doi.org/10.1109/IRASET48871.2020.9092142>
7. Arabi, H., Balakrishnan, V., Mohd Shuib, N.L.: A context-aware personalized hybrid book recommender system. *J. Web Eng.* **19** (2020). <https://doi.org/10.13052/jwe1540-9589.19343>
8. Dien, T.T., Luu, S.H., Thanh-Hai, N., Thai-Nghe, N.: Deep learning with data transformation and factor analysis for student performance prediction. *Int. J. Adv. Comput. Sci. Appl.* **11**(8), 711–721 (2020). <https://doi.org/10.14569/IJACSA.2020.0110886>
9. Dien, T.T., Hoai-Sang, L., Thanh-Hai, N., Thai-Nghe, N.: Course recommendation with deep learning approach. In: Dang, T.K., Küng, J., Takizawa, M., Chung, T.M. (eds.) *FDSE 2020. CCIS*, vol. 1306, pp. 63–77. Springer, Singapore (2020). https://doi.org/10.1007/978-981-33-4370-2_5

10. Zhang, F., Song, J., Peng, S.: Deep Matrix Factorization for Recommender Systems with Missing Data not at Random. *Journal of Physics: Conference Series* 1060, 012001 (2018). <https://doi.org/10.1088/1742-6596/1060/1/012001>
11. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for CTR prediction. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, pp. 1725–1731. AAAI Press (2017). <https://doi.org/10.24963/ijcai.2017/239>
12. Khanal, S.S., Prasad, P.W.C., Alsadoon, A., Maag, A.: A systematic review: machine learning based recommendation systems for e-learning. *Edu. Inf. Technol.* **25**(4), 2635–2664 (2019). <https://doi.org/10.1007/s10639-019-10063-9>
13. Gomez-Uribe, C.A., Hunt, N.: The netflix recommender system: algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.* **6**(4) (2015). <https://doi.org/10.1145/2843948>
14. George, T., Merugu, S.: A scalable collaborative filtering framework based on co-clustering. In: *Fifth IEEE International Conference on Data Mining (ICDM 2005)*, p. 4 (2005)
15. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009). <https://doi.org/10.1109/MC.2009.263>