IMPROVING DEEP MATRIX FACTORIZATION WITH NORMALIZED CROSS ENTROPY LOSS FUNCTION FOR GRAPH-BASED MOOC RECOMMENDATION

Thanh Le, Vinh Vo, Khai Nguyen and Bac Le
Department of Computer Science, Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam

ABSTRACT

Nowadays, with the fast growth of the Internet, the useful role of learning online is getting increasingly popular. MOOC platforms such as Coursera, Edx, Udemy, etc. are attracting many students from all over the world, with thousands of courses constantly continually being opened and updated. This raises the question of how to suggest courses that learners are interested in. To tackle this problem, we apply the Deep matrix Factorization model to the course suggestion along with the improved loss function. The experiment shows that our course recommendation system achieves better NDCG for top K courses than other methods. And the loss function has improved in NDCG measurement compared to the original DMF model.

KEYWORDS

MOOCs, Recommendation, Deep Matrix Factorization, Graph

1. INTRODUCTION

The recommendation system has gone through three decades of development across many fields, and today the most successful models are using deep learning because of having higher accuracy. In the Fourth Industrial Revolution, online learning became an indispensable need for students who yearn for knowledge, so MOOC platforms such as Coursera[1], Edx[2], Khan Academy[3], etc. contain thousands of courses and millions of students. The rapid increase in the number of courses on MOOCs poses a problem of how to choose the right course for learners themselves.

According to our survey, the current research on the recommendation system for MOOCs as well as its quality is not sufficiently high due to the complexity of assessing user knowledge and the continuity in the process. A good recommendation system enables learners to study more effectively. In 2016, (Jdidou & Khaldi, 2016) and in 2018, (Jdidou & Khaldi, 2018) showed that the course recommendation system would optimize students' profitability. (Yanhui et al., 2015) mentioned that 87.3% of the respondents were satisfied with the effectiveness of a course recommendation system. Furthermore, course recommendation systems have been developed to increase the completion rates of students (Labarthe et al., 2016).

Data from MOOCs is often the course name, content, as well as student information, etc. One person can study many different courses. From there, we can simulate in the form of knowledge graphs in which users, courses, etc. will be the vertices of the graph and the edge will be a relationship such as learning, voting, etc. This graph will get bigger and more complex as more courses are created and the number of learners grows. Recommending a suitable course for a specific user will be complicated both in time and accuracy. Because data can be represented in the form of knowledge graphs, we surveyed approaches in knowledge graph mining and combined them with collaborative filtering methods in the recommendation system.

Deep learning has exploded in recent years and has been applied in many fields (Salakhutdinov et al., 2007). In 2007, (Salakhutdinov et al., 2007) proposed a Restricted Boltzmann Machine model called RBM-CF, which

[1] https://www.coursera.org/
[2] https://www.edx.org/
[3] https://www.khanacademy.org/

was the first deep learning model used in the recommender system to bring higher results than the classical method, This model has a simple structure with only one hidden layer. (Elkahky et al., 2015) developed the MV-DNN model by combining multiple Deep Structured Semantic Model (DSSM) models to take advantage of common information across regions. (Kim et al., 2016) integrated convolution matrix factorization into probability matrix factorization (PMF) in the contextual analysis of the learners. (Tan et al., 2016) improved the RNN network by enhancing data to directly predict item embedding. (Y. Wu et al., 2016) suggested a Collaborative Denoising Autoencoder (CDAE) model that was improved from the Denoising AutoEncoder (DAE) model to solve the problem of the recommendation system, and CDAE also integrates human information to achieve greater accuracy.

Later, (Xue et al., 2017) proposed the Deep Matrix Factorization (DMF) method by combining matrix factorization and DMF model, while improving the loss function to achieve better effectiveness. (Wang et al., 2015) presented a new model called Collaborative Deep Learning (CDL) using a hierarchical Bayesian model and Stacked Denoising Autoencoders (SDAE) to solve the sparsity problem of data. (Li et al., 2017) use an autoencoder network to handle implicit and explicit information based on matrix factorization, and at the same time combining supervised learning and unsupervised learning to enhance model efficiency. In 2020, (Pan et al., 2020) upgraded the CDAE model by combining three small CDAE models to a new model called CoDAE (Correlation Denoising Autoencoder), this model was experimented and gave better results than the CDAE model in 2016.

Deep learning also is employed for MOOCs recommender systems, many works applied deep learning methods. (Raghuveer et al., 2014) introduced a reinforcement learning model to generate the learning context and analyze the learner's information. (Mi & Faltings, 2016) offered context trees applied to the online sequential recommendation. (Yang et al., 2014) use matrix factorization and context information forum apply on Forum thread. (Kardan et al., 2017) adopted social network analysis and association rule mining for MOOC forums. (Pardos et al., 2019) operated Recurrent Neural Networks to handle learner's time on each page for predicted courses. (Jing & Tang, 2017) construct a content-awareness framework using users' access information to represent students' interest and behavior features. (Zhang et al., 2017) used a deep belief network for the first time in MOOC recommendation. Then, (Zhang et al., 2019) improved a higher accurate recommendation model using learner's information and content features of the course by using learner-course feature vectors as inputs. While effectively, these methods need data that has a lot of information about courses, user's backgrounds like hobbies, actions, history, private information. This information is difficult to collect and handle. Additionally, these models have very high computation with a lot of calculation time.

Among the related works, we have found that DMF is suitable for the MOOC recommendation system because of its effectiveness. Therefore, in the next section, we present the basic theoretical principles of these methods, and improvements to increase the accuracy of the system.

The remainder of this paper is structured as follows. Part 2 presents the theoretical principle DMF model and our improvements. In Part 3, we conduct experiments and results on the travel-well dataset. Part 4 is the conclusion and our future works.

2. DEEP MATRIX FACTORIZATION

In the recommender system, data includes users and courses will be stored in a matrix. To use this matrix for later problems, we take methods in graph embeddings such as random walk approaches, deep approaches, factorization approaches, etc. Among them, the deep approaches are being applied more popularly. Therefore, we employ the matrix factorization using deep learning because of its effectiveness in our problem.

2.1 Deep Structured Semantic Model

Deep Structured Semantic Model (DSSM) is proposed by (Huang et al., 2013) for web search. Initially, DSSM maps query and documents into lower semantic space with a multi-layer non-linear projection. Then, for ranking webpage, cosine similarity is used.

Specifically, DSSM receives high dimensional vectors (converted from text features) as inputs. It transfers inputs to two multi-layer perceptrons. Then, map them into semantic vectors in a shared semantic space.

Suppose the input is vector **x**, the output is vector **y**, $\mathbf{l_i}$ is the ith hidden layer, $\mathbf{W_i}$ is the ith weight matrix, $\mathbf{b_i}$ is the ith biased. From interaction matrix **Y**, each user $\mathbf{u_i}$ is a vector of $\mathbf{Y_{i*}}$ meaning ith user rates for all items. Each item $\mathbf{v_j}$ is a vector $\mathbf{Y_{*j}}$ meaning jth item rated by all users. Multi-layer perceptrons (MLPs) use (1).

$$l_1 = W_1 x$$

$$l_i = f(W_{i-1} l_{i-1} + b_i), i=2,\ldots N-1 \tag{1}$$

$$y = f(W_N l_{N-1} + b_N)$$

The similarity between the semantics of query and documents uses cosine similarity in (2).

$$R(Q,D) = \text{cosine}(p_i, q_j) = \frac{p_i^T . q_j}{\|p_i\| . \|q_j\|} \tag{2}$$

2.2 Deep Matrix Factorization Model

Deep Matrix Factorization (DMF) is a technique that combines the Matrix Factorization technique (MF) and DSSM. It receives explicit rating and zero implicit feedback and predicts courses based on the correlation of courses. The DMF model takes an interaction matrix. Similar to DSSM, this matrix split into two multi-layer perceptrons (MLPs in (1)). As the result, the output of these MLPs is latent representations. Finally, for calculating the correlation between two latent representations, we calculate cosine similarity. Fig. 1 illustrates DMF architecture.

Given a set include M users: $\mathbf{U=\{u_1,u_2,\ldots,u_M\}}$, and a set include N items: $\mathbf{I=\{i_1,i_2,\ldots,i_N\}}$. $\mathbf{R \in R^{MxN}}$is the rating matrix with $\mathbf{R_{ij}}$ is rating of user **i** for item **j**, **unk** is unknown rating. Equation (3) present the user-item interaction matrix.

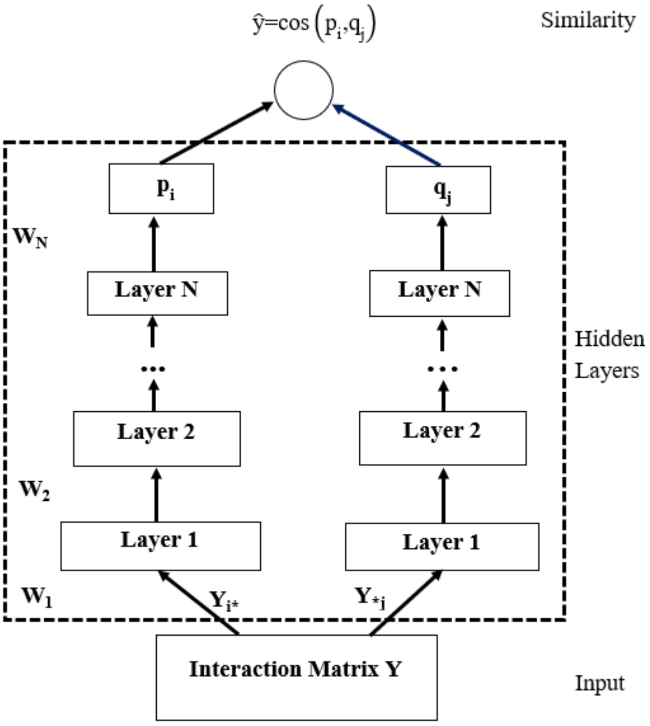

Figure 1. DMF architecture

$$Y_{ij}=\begin{cases} 0,\text{if } R_{ij}=\text{unk} \\ R_{ij},\text{ otherwise} \end{cases} \tag{3}$$

where **u** is user, **v** is item; **i**, **j** is the index of **u**, **v**. **Y** is user-item interaction matrix, $\mathbf{Y}^{+}$is observed interactions, $\mathbf{Y}^{-}$ is zero elements in **Y**, $\mathbf{Y}^{-}_{\mathbf{sampled}}$ is a set of negative instances from Y (in part or in whole). $\mathbf{Y}^{-}\cup\mathbf{Y}_{\mathbf{sampled}}$ is a set of training interactions. Row **i** of matrix **Y** is $\mathbf{Y}_{\mathbf{i}*}$, column **j** of the matrix is $\mathbf{Y}_{*\mathbf{j}}$.

We use ReLU activation function in (4).

$$f(x) = \max(0,x) \tag{4}$$

This model has two MLPs, one for users and one for items, and outputs are mapped into low dimensional vectors in latent space in (5).

$$\begin{gathered} p_i= f_{\theta_N^U}\left(\ldots f_{\theta_3^U}\left(W_{U2}f_{\theta_2^U}(Y_{i*}W_{U1})\right)\ldots\right) \\ q_j= f_{\theta_N^I}\left(\ldots f_{\theta_3^I}\left(W_{V2}f_{\theta_2^I}\left(Y_{*j}^{T}W_{V1}\right)\right)\ldots\right) \end{gathered} \tag{5}$$

At that time, we calculate the cosine similarity of two latent representations p_i and q_j with (2).

In the next part, we will present an improved loss function which increases the accuracy of the model.

2.3 Loss Function

The general objective function in (6).

$$L=\sum\nolimits_{u\in Y^{+}\cup Y^{-}} l(y,\hat{y})+\lambda\Omega(\theta) \tag{6}$$

where, $\Omega(\theta)$ is a regularizer and l(.) is a loss function.

The loss function is an important part of the objective function. The better loss function is, the better the objective function is. Hence, we optimize the objective function by improving the loss function.

Basically, binary cross-entropy is a popular loss function. There are many papers using binary cross-entropy in their works (J. Wu et al., 2009) (Equation (7)).

$$L_{BCE} = -\sum\nolimits_{(i,j)\in Y^{+}\cup Y^{-}} Y_{ij}\log\widehat{Y_{ij}}+\left(1-Y_{ij}\right)\log\left(1-\widehat{Y_{ij}}\right) \tag{7}$$

Equation (7) works effectively with implicit feedback because it considers implicit feedback classification as binary classification. Since both zero implicit feedback and explicit rating are used, we deploy a new loss function by combining binary cross-entropy in (7) with max rating. The $\frac{Y_{ij}}{\max(\text{Rating})}$ is in range [0,1], so it is called Normalized cross-entropy loss (NCE) (Xue et al., 2017) (Equation (8)).

$$L_{NCE} = -\sum\nolimits_{(i,j)\in Y^{+}\cup Y^{-}} \left(\frac{Y_{ij}}{\max(\text{Rating})}\log\widehat{Y_{ij}} + (1-\frac{Y_{ij}}{\max(\text{Rating})})\log\left(1-\widehat{Y_{ij}}\right)\right) \tag{8}$$

In our data, we use max(Rating)=5 because 5 is the max rating.

This model uses direct input as an interaction matrix and is very useful in representing the final low dimensional. Normalized cross-entropy can make the predicted score of $\mathbf{Y}_{\mathbf{ij}}$ be negative so that we use (9) to solve this problem.

$$\widehat{Y_{ij}^{O}}=\max\left(\mu,\hat{Y}_{ij}\right) \tag{9}$$

where $\mu=10^{-6}$.

We represent the DMF with NCE loss function in Algorithm 1.

```
Algorithm 1: NCE_DMF (Iter, neg-ratio, R)
Inputs:
Iter # The number of iterations
neg-ratio #negative ratio
R # Interaction matrix
Outputs:
$W_{Ui}$(i=1…N-1)# weight matrix for user
$W_{Vi}$(i=1…N-1)# weight matrix for item
1. Initialize:
   a. Initialize randomly $W_U$ and $W_V$
   b. Y := Use (3)
   c. $Y^+$:= All non-zero interactions in $Y$;
   d. $Y^-$ :=All zero interactions in $Y$;
   e. $Y^-_{sampled}$ := sample neg _ ratio* $\|Y^+\|$
      (interactions from $Y^-$)
   f. T:=$Y^+ \cup Y^-_{sampled}$
2. Loop it from 1 to $Iter$:
      Loop each interaction of user i and item j in T:
         $p_i, q_j$ := Use (5)
         $\hat{Y}^O_{ij}$ := Use (2) and (9)
         L := Use (8).
      End for
   End for
```

3. EXPERIMENTS AND RESULTS

3.1 Dataset

The Travel-well dataset **Error! Reference source not found.**(Verbert et al., 2011) was collected from the learning resource exchange portal includes 20 content providers from Europe and elsewhere. It contains information about the ratings and tagging behaviors of 98 learners in over six months. Travel-well dataset is used for our experiment because some other datasets are not suitable for our model or not public. In our experiment, we only use rating information of this dataset.

Table 1. Travel-well dataset

#learners (#users)	#courses (#items)	#ratings	density
75	1608	2156	0.0178

3.2 Parameter Settings

We run at following requirements but not limited to: python = 3.7.6, with some libraries Tensorflow-gpu=1.5.0, numpy = 2.1.0.

Parameter settings: learning rates=0.0001, max epoch=50, batch size=256, early stopping = 5, K=1, 5, 10, 20, 30, 50. We choose the best hyper-parameter for the model by trying various parameter settings and then select ones that has the best accuracy. We applied k-fold cross validation for better validation with k = 10.

3.3 Metrics

To evaluate performance, we adopted the *leave-one-out* evaluation. Two metrics, *Normalized Discounted Cumulative Gain* (*NDCG*) in (He et al., 2015; Järvelin & Kekäläinen, 2002), are used to evaluate the ranking performance.

3.4 Results

To validate the effectiveness of our model, we have selected three following models to compare. They are good approaches in recommendation system based on explicit rating.

- **SVD** (Singular value decomposition) (Toroslu, 2010): This approach uses singular value decomposition to reduce the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<<N). It applies a matrix structure where each row represents a user, and each column represents an item, and the elements of this matrix are the ratings that are given to items by users.
- **AutoRec** (Sedhain et al., 2015): AutoRec uses the autoencoder paradigm to design an item-based (user-based) by reconstructing the partially observed vectors.
- **DMF** (Xue et al., 2017): This model combines matrix factorization and deep structured semantic models.

Table 2 presents the detailed results on the NDCG@K metric with K = [1, 5, 10, 20, 30, 50] as in the original DMF, AutoRec, SVD and NCE_DMF models. The NCE-DMF shows better performance compare to SVD, AutoRec, and DMF models respectively. It is surprising with the NDCG@K of AutoRec model (<1% for all different K).

Table 2. NDCG@K with K = [1, 5, 10, 20, 30, 50] of the DMF, AutoRec, SVD, and NCE-DMF

K	DMF	AutoRec	SVD	NCE_DMF
1	0.3467	0.0019	0.4927	**0.5936**
5	0.3945	0.0039	0.4875	**0.7412**
10	0.4701	0.0040	0.4833	**0.7780**
20	0.5000	0.0043	0.4800	**0.7698**
30	0.5493	0.0059	0.4789	**0.7770**
50	0.5762	0.0070	0.4770	**0.6826**

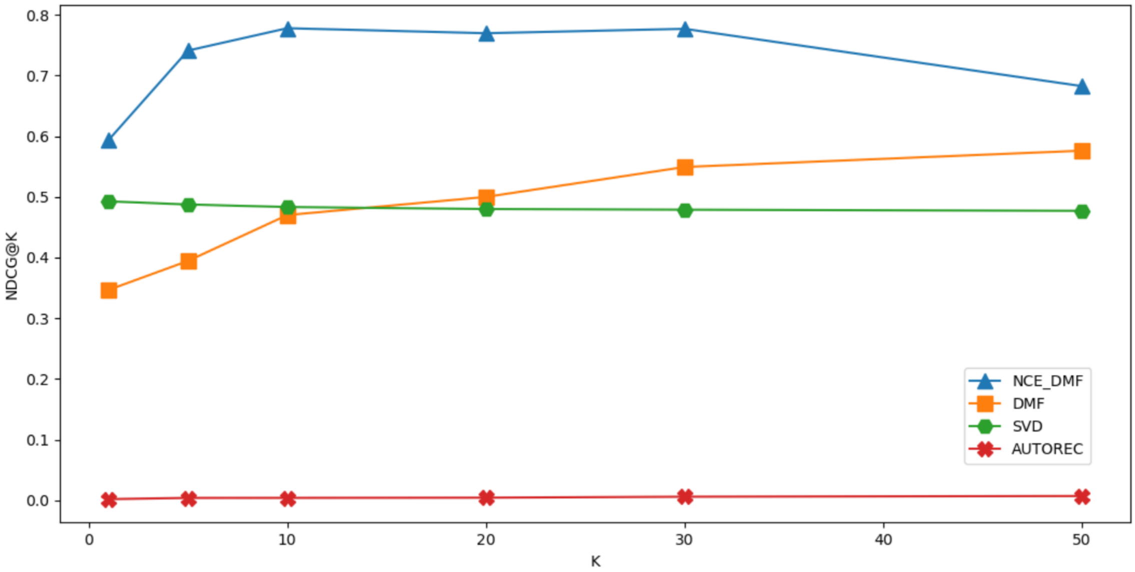

Figure 2. Comparison with NDCG@K on the Travel-well dataset

Figure 2 illustrates the NDCG@K metrics of the NCE_DMF, AutoRec, SVD and DMF models. In this chart, the AutoRec and DMF increase NDCG when K get bigger. However, the NDCG of SVD decreases when K increases. For NCE_DMF model, NDCG increases when $K \leq 30$, but decreases when K=50. The cause may come from the uneven distribution of data. Overall, the NCE_DMF model has higher NDCG ranking results than other related methods in the MOOC recommender system. This shows that we can apply NCE_DMF to improve the quality of the MOOC recommendation system.

4. CONCLUSION

In this paper, we improved the DMF model with a new loss function and applied it to the MOOC recommendation system. The experiment shows that the proposed approach is better than the other models on NDCG@K measurement. Specifically, our algorithm improved an average 25.09% compared to the DMF model, and 24.04% compared to SVD method. In the future, we will continue to improve DMF with some other loss functions. Moreover, this model can be expanded from zero of implicit feedback to implicit feedback containing user feedback.

ACKNOWLEDGEMENT

This research is funded by University of Science, VNU-HCM, under grant number CNTT 2020-05.

REFERENCES

Elkahky, A. M., Song, Y., & He, X. (2015). A multi-view deep learning approach for cross domain user modeling in recommendation systems. *Proceedings of the 24th International Conference on World Wide Web*, 278–288.

He, X., Chen, T., Kan, M.-Y., & Chen, X. (2015). Trirank: Review-aware explainable recommendation by modeling aspects. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1661–1670.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, 2333–2338.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, *20*(4), 422–446.

Jdidou, Y., & Khaldi, M. (2016). Increasing the Profitability of Students in MOOCs using Recommendation Systems. *International Journal of Knowledge Society Research (IJKSR)*, *7*(4), 75–85.

Jdidou, Y., & Khaldi, M. (2018). Using Recommendation Systems in MOOC: An Innovation in Education That Increases the Profitability of Students. In *Enhancing Knowledge Discovery and Innovation in the Digital Era* (pp. 176–190). IGI Global.

Jing, X., & Tang, J. (2017). Guess you like: course recommendation in MOOCs. *Proceedings of the International Conference on Web Intelligence*, 783–789.

Kardan, A., Narimani, A., & Ataiefard, F. (2017). A hybrid approach for thread recommendation in MOOC forums. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, *11*(10), 2195–2201.

Kim, D., Park, C., Oh, J., Lee, S., & Yu, H. (2016). Convolutional matrix factorization for document context-aware recommendation. *Proceedings of the 10th ACM Conference on Recommender Systems*, 233–240.

Labarthe, H., Bachelet, R., Bouchet, F., & Yacef, K. (2016). Increasing MOOC completion rates through social interactions: a recommendation system. *Research Track*, 471.

Li, Q., Zheng, X., & Wu, X. (2017). Collaborative autoencoder for recommender systems. *ArXiv E-Prints.*

Mi, F., & Faltings, B. (2016). Adaptive Sequential Recommendation Using Context Trees. *IJCAI*, 4018–4019.

Pan, Y., He, F., & Yu, H. (2020). A correlative denoising autoencoder to model social influence for top-N recommender system. *Frontiers of Computer Science*, *14*(3), 143301.

Pardos, Z. A., Fan, Z., & Jiang, W. (2019). Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, *29*(2), 487–525.

Raghuveer, V. R., Tripathy, B. K., Singh, T., & Khanna, S. (2014). Reinforcement learning approach towards effective content recommendation in MOOC environments. *2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)*, 285–289.

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. *Proceedings of the 24th International Conference on Machine Learning*, 791–798.

Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. *Proceedings of the 24th International Conference on World Wide Web*, 111–112.

Tan, Y. K., Xu, X., & Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 17–22.

Toroslu, Ġ. H. (2010). A singular value decomposition approach for recommendation systems. *The Graduate School of Natural and Applied Sciences of Middle East Technical University, Ph. D. Dissertation*.

Verbert, K., Drachsler, H., Manouselis, N., Wolpers, M., Vuorikari, R., & Duval, E. (2011). Dataset-driven research for improving recommender systems for learning. *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 44–53.

Wang, H., Wang, N., & Yeung, D.-Y. (2015). Collaborative deep learning for recommender systems. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1235–1244.

Wu, J., Wang, Z., & Vojcic, B. R. (2009). Partial iterative decoding for binary turbo codes via cross-entropy based bit selection. *IEEE Transactions on Communications*, *57*(11), 3298–3306.

Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016). Collaborative denoising auto-encoders for top-n recommender systems. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 153–162.

Xue, H.-J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017). Deep Matrix Factorization Models for Recommender Systems. *IJCAI*, *17*, 3203–3209.

Yang, D., Piergallini, M., Howley, I., & Rose, C. (2014). Forum thread recommendation for massive open online courses. *Educational Data Mining 2014*.

Yanhui, D., Dequan, W., Yongxin, Z., & Lin, L. (2015). A group recommender system for online course study. *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, 318–320.

Zhang, H., Huang, T., Lv, Z., Liu, S., & Yang, H. (2019). MOOCRC: A highly accurate resource recommendation model for use in MOOC environments. *Mobile Networks and Applications*, *24*(1), 34–46.

Zhang, H., Yang, H., Huang, T., & Zhan, G. (2017). DBNCF: Personalized courses recommendation system based on DBN in MOOC environment. *2017 International Symposium on Educational Technology (ISET)*, 106–108.