SynCF: A Synthetic Framework for Deep-learning Based Collaborative Filtering

Cong Lin[1]
[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
linc7@mail2.sysu.edu.cn

Abstract

Deep learning methods have been employed in collaborative filtering recommender systems to address many issues, i.e., representation learning and matching function learning, capturing low-order and high-order feature interactions, and combining linearity and non-linearity. However, those issues are mainly studied in separate literature and solved by different models, which means there is no uniform framework that balances the solutions to all those issues. To solve this problem, some frameworks have attempted to deal with several issues at the same time by introducing multiple modules in the model for each target issue, such as the DeepCF by SYSU paralleling representation learning and matching function learning and the PNN by SJTU binding low-order linear interactions and high-order non-linear interactions. Though progress has been achieved, they still possess some flaws in their architecture and cannot cover the rest issues. Based on previous ideas, in this paper we devise a novel framework termed SynCF for deep-learning based collaborative filtering recommender system to synthesize the concerns and solutions on all the six issues mentioned above. We study the case of interaction prediction with implicit feedback data, and design a specific model called Synthetic Collaborative Filtering Network (SCFNet) under SynCF framework. Our rudimentary results from experiences on two datasets catch up the performance by the state-of-the-arts, indicating the effectiveness of our framework.

Keywords: Recommender System, Collaborative Filtering, Deep Learning, Neural Network, Implicit Feedbacks

Introduction

Since the beginning of information explosion era, recommender systems have been developed and rising in the role of tackling information overload problems and offering more precise information for users in various scenarios [1, 2], i.e., online advertising, electronic commerce and social platform. Among approaches to implement a recommender system, the collaborative filtering method has been widely deployed due to its abilities to offer personalized recommendations by also making use of interpersonal preference [3].

The key idea in collaborative filtering recommender systems is that the preference of a specific user on some items is evaluated through the preference of many other similar users. Following this idea, in Matrix Factorization method, a prevailing collaborative filtering model [4, 5, 6], the preprocessed user and item data are represented as embedding vectors of latent factors, and the model makes preference evaluation by a mapping that measures the similarity between the embedding vectors. These two part are called representation part and matching function part respectively [7], and a variety of deep-learning based extensions [8, 9, 10, 11, 12, 13, 14, 15, 16] have been employed to replace the original linear embedding or inner-product based matching function with a more expressive function learned by neural networks. However, many deep neural network extensions only address a single aspect among representation learning [3, 12] and matching function learning [9, 10, 11, 13, 14, 15], resulting in a Deep+Shallow pattern framework that limits their performance. To seek for better expressiveness, DeepCF [7] first points out the significance of incorporating representation learning and matching function learning in deep-learning based collaborative filtering model, but its parallel architecture which completely separates the two learning tasks inherits the weakness of both parts, and still adopts some operation not much better than inner-product.

Besides concerns on combining representation learning and matching function learning, studies have also taken insights into different aspects of matching function design, such as low-order v.s. high-order feature interactions capturing [14] and linear v.s. non-linear mapping [13]. The order of feature interactions refers to the degree to which inter-feature combination is utilized for the matching function part. In recent years, most of state-of-the-arts

in deep-learning based collaborative filtering pay attention to elaborating a matching function part that balances between low-level and high-level feature interactions capturing [9, 13, 15]. But the deficiency they share in common is that the inner-product or element-wise product adopted can only capture an incomplete second-order interaction. The Product-based Neural Network (PNN) [14] addresses this problem by replacing the inner-product with outer-product of vectors so that the second-order interactions between any two feature fields can be captured. But somehow for the sake of convenience, the model takes summation to combine its one-order and second-order interactions and also to reduce the operation in outer-product, probably leading to great information loss. In terms of the balance between linear and non-linear mapping, a relative good conclusion has been drawn that deep neural networks can approximate any continuous functions [16], so if linear mapping outperforms non-linear mapping, it would be learned by the network. Despite this, models like Wide & Deep [9] and Neural Collaborative Filtering (NCF) [13] also explicitly introduce both linear and non-linear part in the framework, guaranteeing that linear mapping would be reserved even under limited network capacity.

To sum up the discussion above, we now have three pairs of complementary properties, namely representation learning v.s. matching function learning [7], low-order v.s. high-order feature interactions capturing [14], and linear v.s. non-linear mapping [13]. All these three pairs are studied in different literature, and there is no uniform framework that is aimed at incorporating and balancing solutions to them. Therefore, we now propose the framework called Synthetic Collaborative Filtering (SynCF) to consider all these six aspects at the same time. The architecture of the framework is built with a representation learning part and a matching function learning part, but they are combined in a series connection way rather than the parallel way in DeepCF, which means that during training phase, the representation part receives better guidance and the matching function can utilized the excavated latent factors more efficiently. Further, the matching function is synthesized with multiple modules that each possess certain properties expected above. The final preference evaluation, also called the matching score, is made by a mapping from the concatenation of the outputs of matching function modules. We study and demonstrate the performance of SynCF on implicit feedback datasets, a common scenario of great importance in practice.

The main contributions of this work are as follows.

- We summarize six significant aspects in collaborative filtering, point out the significance of incorporating them, and propose a general Synthetic Collaborative Filtering (SynCF) framework. The proposed framework is supposed to obtain the strengths of those different properties.
- We design a novel model named Synthetic Collaborative Filtering Network (SCFNet) under the SynCF framework to tackle the recommendation problem with implicit feedback data.
- We conduct rudimentary experiments on two real-world datasets to demonstrate the performance of SynCF and SCFNet.

Related Work

Collaborative Filtering Augmented on Representation and Matching Function

Mainly based on the Matrix Factorization method, deep-learning based collaborative filtering models have long been trying to replace its linear embedding parts and mappings with neural network mappings, which are justified to have powerful expressiveness of approximating any continuous functions given appropriate capacity [16]. On representation learning, characteristic improvement is offered by Deep Matrix Factorization (DMF) [3] which uses two Multi-Layer Perception (MLP) models to learn to embedding mapping from raw user-item data to the latent factor space. Similarly, NeuMF under the Neural Collaborative Filtering (NCF) [13] framework introduces MLP to the matching function part. The core architectures of these two models are combined in a paralleling way in DeepCF model [7], which claims to be the pioneer in incorporating representation learning and matching function learning. However, besides still resorting to dot-product-like element-wise product, its architecture cannot sufficiently utilize the strengths of the two learning components. In our proposed model, the two components are combined in a series connection way so that each of them can take advantage of the other.

Collaborative Filtering Using Product Operations

Since Matrix Factorization method takes inner-product operation as the mapping to interaction score, many deep-learning based collaborative filtering models have also followed up with this design [7, 12, 13], which can turn out to be a flaw because neural networks can build up more complex mappings. A simple augmentation is to concatenate the latent feature vectors of user and item as input to a MLP, which can be found in works such as FNN [11]. Since in concatenation, different features are only linearly combines, the MLP matching function will only capture low-order feature interactions. Both the weaknesses of placing a

concatenation or using element-wise product after the embedding have been pointed out as lacking of modeling correlations between features in Outer product based Neural Collaborative Filtering (ONCF) framework [10], which first uses convolutional mapping function that is not general yet. To our concern, Outer Product-based Neural Network under (OPNN) under Product-based Neural Network (PNN) [14] framework employs outer-product of vectors to capture the second-order interactions between any two latent feature fields while still reserves the one-order learning component. The key problem is that it uses extra summation to combine low-order interactons and high-order interactions and to simplify the outer product, which wipes out some feature information. We will adopt outer-product in our designed SCFNet model by a concatenation approach to avoid information loss while fast operate tensor outer-product.

Collaborative Filtering Combining Linearity and Non-linearity

Despite the fact that neural networks can actually learn linear mappings, Wide & Deep [9] model and NCF [13] model still keep a relatively linear mapping component in their architecture. The Wide & Deep comprises a Deep component and a Wide component, where the later is basically a linear transform on the raw feature input vector. Likewise, NCF contains a swallow Matrix Factorization model as its sub model to ensure that its performance is at least above linear MF model. In our SCFNet model, we will not explicitly use a linear mapping component. Instead, we leave it for the deep NCF-like component to learn if linearity indeed turns out to be optimal.

Preliminaries

Problem Statement

We follow the problem statement of DeepCF [7], which is also inherited from [13, 17]. That is, for a user-item system of M users and N items, we construct a user-item interaction matrix $\mathbf{Y} \in \mathbb{R}^{M \times N}$as

$$y_{ui} = \begin{cases} 1, & \text{if interaction (user } u\text{, item } i\text{) is observed;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

and we are required to estimate the missing values in interaction matrix $\mathbf{Y}$, namely estimating whether unobserved interaction would happen or not, which is also known as the One-Class Collaborative Filtering (OCCF) problem [18].

While solving this issue in a discrete and binary way does not help the recommender system with ranking and recommending items, a widely adopted solution is to assume that y_{ui} follows a Bernoulli distribution:

$$\begin{aligned} P(y_{ui} = k|p_{ui}) &= \begin{cases} 1 - p_{ui}, & k = 0; \\ p_{ui}, & k = 1 \end{cases} \\ &= p_{ui}^k (1 - p_{ui})^{1-k}, \end{aligned} \tag{2}$$

where p_{ui} can either be considered as the probability of y_{ui} being equal to 1, or intuitively interpreted as the probability that user u shows interest on item i.

By introducing p_{ui}, our target to model becomes p_{ui} rather than y_{ui}. The original discrete and binary classification problem is now turned into a continuous matching score prediction problem, which can be solved much more easily.

Learning the Model

In general, deep-learning based collaborative filtering models aim at training a model f with trainable parameters Θ such that given specific user u and item i, $f(u, i|\Theta)$ yields $\hat{y}_{ui}$ as the prediction of y_{ui}, namely p_{ui} the predicted probability that user u is matched by item i. In deep learning, the parameters are simply updated by backpropagaion to optimize the objective loss function. Two types of loss functions are commonly used in recommender system, the point-wise loss [19] and pair-wise loss [20]. In this paper, we choose point-wise loss following [7], and specifically use the binary cross-entropy loss function:

$$\ell_{BCE} = -\sum_{(u,i) \in \mathcal{Y}^+ \cup \mathcal{Y}^-} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}). \tag{3}$$

where $\mathcal{Y}^+$ denotes all the observed interactions in $\mathbf{Y}$ and $\mathcal{Y}^-$ denotes the unobserved interactions sampled as negative instances, i.e., the negative instances, as is done by [18].

Now we have already defined the problem and answered how to estimate parameters in the neural network model. In the next section, we will define our neural network framework.

The Proposed Model

We first introduce the general architecture under the proposed SynCF framework, and then exemplify it for the SCFNet.

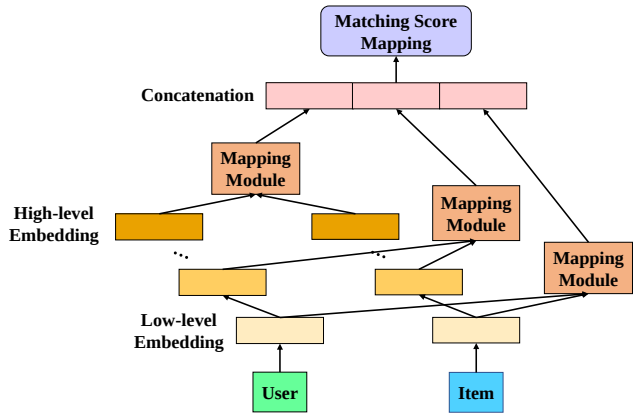

Figure 1: The general architecture under SynCF framework. To some extent, it is built in a sideout manner that mapping modules can be attached to the suitable embedding layer.

The SynCF Framework

The most crucial weakness of DeepCF [7] which tries to incorporate representation learning and matching function learning, is that the two components are combined in such a paralleling way that the strengths of each of them cannot be shared by the other and still retains the weaknesses. To avoid this, we make the matching function learning component follow up the the representation learning component so that driven by a more expressive matching function, the representation learning component would extract latent factors more precisely, and that receiving ampler latent factors, the expressiveness of matching function learning component would be fully exploited.

For the rest four issues, low-order and high-order feature interactions capturing and linear and non-linear mapping, modules responsible for them can be incorporated in the matching function learning component. We further argue that unlike directly joining all kinds of modules together, which obviously makes no sense and can lead to terrible performance, we incorporate them more flexibly in such a way that different mapping function modules can be attached to different levels of representation learning, as is shown in Figure 1. To get the final matching score on the user and item, the outputs of mapping function modules are concatenated as the input to the final mapping. In this way, the final mapping does not need to be a complex MLP model since previous mapping modules are sufficiently expressive.

The SCFNet Model

As is shown in Figure 2., our proposed SCFNet uses MLPs as the embedding layers and MLP-based mapping modules to build up the matching function. We use 2 separated MLPs for the representation learning of user and item. Here, we denote $\boldsymbol{y}_{u*}$ as the input raw user rating vector and $\boldsymbol{y}_{*i}$ as the input raw item rating vector. Taking the representation part for users as example, it is defined as:

$$\begin{aligned} \mathbf{embed}_1^{\boldsymbol{u}} &= \mathbf{W}_1^T \boldsymbol{y}_{u*} \\ \mathbf{embed}_2^{\boldsymbol{u}} &= a(\mathbf{W}_2^T \mathbf{embed}_1^{\boldsymbol{u}} + \mathbf{b}_1) \\ &\cdots\cdots \\ \mathbf{embed}_k^{\boldsymbol{u}} &= a(\mathbf{W}_k^T \mathbf{embed}_{k-1}^{\boldsymbol{u}} + \mathbf{b}_{k-1}), \end{aligned} \tag{4}$$

where $\mathbf{embed}_k^{\boldsymbol{u}}$ is the k^{th} embedding vector of user vector $\boldsymbol{y}_{u*}$, a is the activation function which is usually Relu, and $\mathbf{W}_k$ and $\mathbf{b}_k$ is the weight parameters in an MLP layer. The case for items is similar.

Two mapping function modules are used in the SCFNet. As is shown in Figure 2., the module taking higher level embedding results is called Product-based Matrix Factorization (PMF) module, and the other module is called NCF module following Neural Collaborative Filtering [13]. PMF is basically a general improved version of DMF [3] where inner-product is replaced and generalized by outer-product. It is responsible for capturing high-order feature interactions. Since NCF is a efficient model for low-order feature interaction capturing and non-linear mapping (covering linear mapping) as is also shown in DeepCF architecture, we simply adopt it here except that it takes MLP embedding vectors instead of linear embedding vectors.

For the details of the PMF module, it can be formalized as:

$$\begin{aligned} \mathbf{product} &= \mathbf{embed}_2^{\boldsymbol{u}} \times \mathbf{embed}_2^{\boldsymbol{i}} \\ \mathbf{prediction}_{\boldsymbol{PMF}} &= MLP(\mathbf{product}) \end{aligned} \tag{5}$$

where $\mathbf{embed}_2^{\boldsymbol{u}}$ and $\mathbf{embed}_2^{\boldsymbol{i}}$ are the last embedding vector for user and item (in our implementation it is the second layer), $\mathbf{product}$ is the outer product result, MLP stands for the MLP part in this module, and $\mathbf{prediction}_{\boldsymbol{PMF}}$ is its output vector.

Since in practice, the batch of $\mathbf{embed}_2^{\boldsymbol{u}}$ or $\mathbf{embed}_2^{\boldsymbol{i}}$ is a 3-dimensional tensor, it will be ambiguous to define the outer-product of them. To avoid this problem and accelerate the calculation, we multiply $\mathbf{embed}_2^{\boldsymbol{u}}$ and each element in $\mathbf{embed}_2^{\boldsymbol{i}}$, and concatenate all the resulted vector. The concatenated vector contain all elements in the outer product matrix, which are the second-order feature interactions we are looking for.

After PMF module produces $\mathbf{prediction}_{\boldsymbol{PMF}}$ and NCF

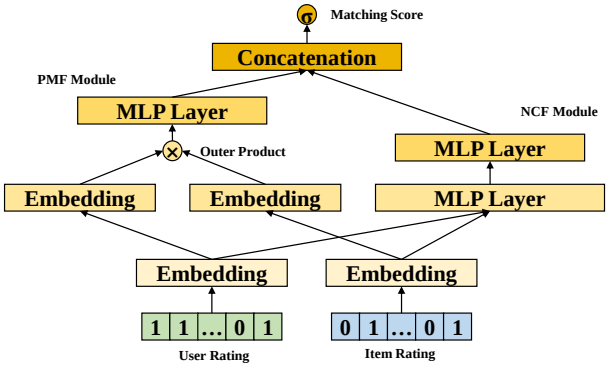

Figure 2: The architecture of SCFNet.

Table 1: Statistics of the Datasets.

Statistics	ml-1m	AMusic
# of Users	6040	1776
# of Items	3706	12929
# of Ratings	1000209	46087
Sparsity	0.9553	0.9980

module similarly produces $\mathbf{prediction}_{NCF}$, in the last stage, we concatenate these two prediction vector, use a one layer MLP to transform it into a scalar, and take a sigmoid function to obtain the final matching score.

Learning We mainly follow the learning procedure of DeepCF [7] except that pre-training is not taken and we train the model using Adam optimizer directly.

Experiments

We now conduct rudimentary experiments on two real-lift datasets to demonstrate the effectiveness of the proposed SynCF framework and SCFNet model. Hype-parameter sensitivity on the architecture of the SCFNet is also analysed. The model is implemented based on Keras[1] with Tensorflow[2] as the backend.

Experimental Settings

Dataset The two real-life datasets to test on are: MovieLens 1M (ml-1m)[3], and Amazon music (AMusic)[4]. The former dataset has been prepocessed by the provider, and we use a version of the latter that is prepocessed in the same way and provided by DeepCF [7]. Summary of the two datatsets can be found in Table 1.

[1] https://github.com/keras-team/keras
[2] https://github.com/tensorflow/tensorflow
[3] https://grouplens.org/datasets/movielens/
[4] http://jmcauley.ucsd.edu/data/amazon/

Evaluation Protocols We follow the evaluation approaches by [7, 13], that is to adopt the leave-one-out evaluation. In details, we first randomly sample 100 unobserved interactions for each user and then rank them with test item according to the matching scores. Hit Ratio (HR) and Normalized Discounted Cumulative Gain are taken as evaluation measures after that. The former evaluates whether test item appears to be among the top 10 candidates, while the latter will assign higher scores for precisely hitting to measure the ranking quality.

Results

We compares our model with the methods following.

- **ItemPop** is an often used a benchmark for recommendation tasks.
- **eALS** [21] is an advanced MF method using all unobserved interactions as negative instances.
- **DMF** [3] is a state-of-the-art representation learning-based MF method using MLP for embedding.
- **NeuMF** [13] is a state-of-the-art matching function learning-based MF method which uses MLP for matching score prediction and combines linearity and non-linearity.
- **CFNet** [7] is a state-of-the-art deep-learning based MF method that combines representation learning and matching function learning by using DMF and NeuMF under DeepCF framework.

The comparison results are listed in Table 2. We can draw some basic conclusions from observations:

- Our proposed SCFNet model under SynCF framework catches up with the performance of state-of-the-art models such as NeuMF and CFNet, but there is still a gap around 1 percentage to go to exceed CFNet under DeepCF framework. Since we only conduct rudimentary experiments on the configuration of SCFNet, these results basically demonstrate the effectiveness of our model, and it is believed that better outcomes will be achieved after elaborating other configuration.
- Our model greatly outperforms DMF but not very distinguishable from NeuMF. In our proposed model, we use PMF which basically repleces the inner product in DMF with outer product as a mapping module for capturing high-order feature interactions. Hence, it is reasonable to find that our model exceeds DMF a lot. As for the case of NeuMF, we only adopt its non-linear NCF part in our SCFNet as a mapping module for capturing low-order feature interactions, which may turn out to be not sufficient in rebuilding linear mapping as is explicitly done by NeuMF.

Table 2: Comparison results of different methods in terms of NDCG@10 and HR@10.

Datasets	Measures	Existing methods					SynCF
		ItemPop	eALS	DMF	NeuMF	CFNet	SCFNet
ml-1m	HR	0.4535	0.7018	0.6565	0.7210	0.7253	0.7152
	NDCG	0.2542	0.4280	0.3761	0.4387	0.4416	0.4373
AMusic	HR	0.2483	0.3711	0.3744	0.3891	0.4116	0.4099
	NDCG	0.1304	0.2352	0.2149	0.2391	0.2601	0.2427

Table 3: Performance of SCFNet on ml-1m with different configuration of embedding and PMF.

Embedding	PMF	HR	NDCG
256-64	1	0.7031	0.4194
256-64	32-1	0.7066	0.4285
256-64	64-1	0.7104	0.4313
256-64	128-1	0.7152	0.4373
256-64	256-1	0.7103	0.4355
256-64	64-16-1	0.7116	0.4286
256-64	128-64-1	0.7055	0.4281
256-64	256-64-1	0.7096	0.4333
1024-64	64-1	0.7043	0.4278
512-64	64-1	0.7051	0.4304
128-64	64-1	0.7098	0.4314

This indicates that improvements of our model can be further made in terms of linear mapping.

Sensitivity Analysis of Architecture

We fix the architecture configuration of NCF module which has been justified by DeepCF and NeuMF, and mainly analyze the architecture of the newly proposed PMF module on ml-1m dataset. The configuration of the representation part is also tested. Results are shown in Table 3. The optimal performance is provided by architecture with an 2 layers embedding part that capture 256 and 64 features respectively, and an 2 layers PMF model that will mapping the outer-product of 64×64 to size of 128 and the for the matching score scalar. This observation shows that both the embedding part and PMF module don't need to be very deep and wide, so the improvement by the proposed model is not strongly made by the capacity but still made by the framework.

Conclusion and Future Work

We have studied previous works on deep-learning based collaborative filtering recommender system in terms of six crucial aspects, namely represenation learning, matching function learning, low-order feature interaction capturing, high-order feature interaction capturing, linear mapping and non-linear mapping. These six aspects are mainly studied in different literature and partly solved by different models. Only a few works have attempted to incorporate some aspects among them, which is still a long way to go. As a result of this observation, we point out the significance of incorporating all these 6 crucial aspects in a uniform framework. In this paper, we first propose the general framework SynCF for combining the six aspects, and then design a novel model called SCFNet under SynCF framework. The SynCF framework is special in that it highlights the way that representation learning part and matching function learning part are joined, and devises a synthetic matching function using different mapping function modules. The SCFNet is implemented basically with NCF and outer-product based extension to DMF. The proposed framework and model have shown rudimentary results indicating their advanced performance, but there is still a little gap between the state-of-the-art, which is worth further researching. In future work, following questions can be studied to further improve the model. First, other mapping module can replace PMD and NCF in our propose SCFNet for better high-order feature interactions capturing and both linear and non-linear mapping. Second, the configuration of the architecture can be more carefully studied to find the optimal depth and width of the neural network. Third, unlike DeepCF, pretraining does not work well with our proposed model since its representation learning component and matching function learning component are merged together rather than paralleling. Thus, training approaches such as pretraining or training with phases can be studied and adopted in the future version of SCFNet. Finally, other attempts such as using pair-wise loss, explicit data and other loss function can also be studied for better performance in different scenario.

References

[1] Hu Q Y, Huang L, Wang C D, et al. Item orientated recommendation by multi-view intact space learning with overlapping[J]. Knowledge-Based Systems, 2019, 164: 358-370.

[2] Srivastava R, Palshikar G K, Chaurasia S, et al. Whats Next? A Recommendation System for Industrial Training[J]. Data Science and Engineering, 2018, 3(3): 232-247.

[3] Xue H J, Dai X, Zhang J, et al. Deep Matrix Factorization

Models for Recommender Systems[C] IJCAI. 2017: 3203-3209.

[4] Wang C D, Deng Z H, Lai J H, et al. Serendipitous recommendation in e-commerce using innovator-based collaborative filtering[J]. IEEE transactions on cybernetics, 2018, 49(7): 2678-2692.

[5] Zhao Z L, Huang L, Wang C D, et al. Low-rank and sparse cross-domain recommendation algorithm[C] International Conference on Database Systems for Advanced Applications. Springer, Cham, 2018: 150-157.

[6] Hu Q Y, Zhao Z L, Wang C D, et al. An item orientated recommendation algorithm from the multi-view perspective[J]. Neurocomputing, 2017, 269: 261-272.

[7] Deng Z H, Huang L, Wang C D, et al. Deepcf: A unified framework of representation learning and matching function learning in recommender system[J]. arXiv preprint arXiv:1901.04704, 2019.

[8] Bai T, Wen J R, Zhang J, et al. A neural collaborative filtering model with interaction-based neighborhood[C] Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 2017: 1979-1982.

[9] Cheng H T, Koc L, Harmsen J, et al. Wide deep learning for recommender systems[C] Proceedings of the 1st workshop on deep learning for recommender systems. ACM, 2016: 7-10.

[10] He X, Du X, Wang X, et al. Outer product-based neural collaborative filtering[J]. arXiv preprint arXiv:1808.03912, 2018.

[11] Zhang W, Du T, Wang J. Deep learning over multi-field categorical data[C] European conference on information retrieval. Springer, Cham, 2016: 45-57.

[12] He X, Chua T S. Neural factorization machines for sparse predictive analytics[C] Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2017: 355-364.

[13] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C] Proceedings of the 26th international conference on world wide web. International World Wide Web Conferences Steering Committee, 2017: 173-182.

[14] Qu Y, Cai H, Ren K, et al. Product-based neural networks for user response prediction[C] 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016: 1149-1154.

[15] Guo H, Tang R, Ye Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction[J]. arXiv preprint arXiv:1703.04247, 2017.

[16] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural networks, 1989, 2(5): 359-366.

[17] Wu Y, DuBois C, Zheng A X, et al. Collaborative denoising auto-encoders for top-n recommender systems[C] Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 2016: 153-162.

[18] Pan R, Zhou Y, Cao B, et al. One-class collaborative filtering[C] 2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008: 502-511.

[19] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets[C] 2008 Eighth IEEE International Conference on Data Mining. Ieee, 2008: 263-272.

[20] Mnih A, Teh Y W. Learning label trees for probabilistic modelling of implicit feedback[C] Advances in Neural Information Processing Systems. 2012: 2816-2824.

[21] He X, Zhang H, Kan M Y, et al. Fast matrix factorization for online recommendation with implicit feedback[C] Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2016: 549-558.