An Effective Implicit Multi-interest Interaction Network for Recommendation

Wei Yang[1,2], Xinxin Fan[3], Yiqun Chen[1,2], Feimo Li[1,2](✉), and Hongxing Chang[1,2]

[1] University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Automation, Chinese Academy of Sciences, Beijing, China
{yangwei2019,chenyiqun2020,lifeimo2012,hongxing.chang}@ia.ac.cn
[3] Beihang University, Beijing, China
zf1921334@buaa.edu.cn

Abstract. Data features in real industrial recommendation scenarios are high-dimensional, diverse and sparse. Rich feature interaction can improve the model effect and bring practical benefits. Factorization machines (FMs) can perform explicit second-order feature interactions, while deep neural networks (DNNs) can perform implicit non-linear feature interactions. A series of models integrating FMs and DNNs are used to perform diverse feature interactions. However, most of the previous work performed feature interaction without considering the diverse interests of users. In reality, users often have multiple preferences and interests, which are implicitly included in the features and need to be effectively extracted. In this paper, we propose an implicit multiple interest network (IMIN), taking into account the importance of interest. Specifically, the model constructs the implicit multiple interests of the user and the item through the implicit multi-interest layer, and realizes the interest alignment between the user and the item through the interest alignment layer. We further use the interest interaction and aggregation layer to construct rich interest feature interactions. In addition, we introduce an auxiliary loss in the model optimization part to ensure the difference of interest. We conducted comprehensive and rich experiments on three real-world data sets. Experimental results show that IMIN performs better than other competitive models, which proves the effectiveness of the model.

Keywords: Multiple interest · Feature interaction · Recommendation

1 Introduction

With the explosion of Internet information, recommendation systems play an important role in information matching [2], which are used in media, entertainment, e-commerce and other scenarios [16]. At the same time, the number of users and items increases exponentially, which greatly increases the difficulty of accurate recommendation. Under the scenario of large-scale recommendation system, there are rich and diverse features [6], including user attribute features, item attribute features, user history features, text features, etc. These features

T. Mantoro et al. (Eds.): ICONIP 2021, LNCS 13111, pp. 680–691, 2021.
https://doi.org/10.1007/978-3-030-92273-3_56

tend to be high dimensional, sparse, and diverse [18]. How to extract effective information from massive data features to provide more accurate recommendation effect is a key problem [24].

With the rise of deep learning [12], deep neural networks (DNNs) have been widely used in natural language processing, recommendation, computer vision and other fields due to its strong nonlinear fitting ability [4]. DNNs have been proved to be able to fit arbitrary functions, so it can be used for complex nonlinear feature combination. However, the implicit learning feature interaction information of DNNs is not always effective. Therefore, many fusion models use FMs to learn explicit low-order feature interaction and DNNs to learn implicit feature interaction [5,7].

However, all previous work concatenates the embedding of all features, and then uses Factorization Machine (FM) or DNNs to learn the interaction between features [3,14,20], without considering the diverse interests of users. In actual situations, users' preferences are often diverse, which means that users have a variety of different interests. For example, the user may like fashionable and luxurious items in terms of dressing, but prefer high-quality and inexpensive items in life. The multiple interests of the user can more fully reflect the characteristics of the user, which needs to be effectively modeled.

Considering the problems mentioned above, we propose an implicit multiple interest network (IMIN), taking into account the importance of interest. Specifically, the model constructs the implicit multiple interests of the user and the item through the implicit multi-interest layer, and realizes the interest alignment between the user and the item through the interest alignment layer. We further use the interest interaction and aggregation layer to construct rich interest feature interactions. In addition, we introduce an auxiliary loss in the model optimization part to ensure the difference of interest. The main contributions of this work can be summarized as follows:

- We propose an implicit multiple interest representation and interest alignment layer, which can not only construct multiple interests of users and items, but also model the matching relationship between interests.
- We propose an implicit multiple interest network (IMIN), taking into account the importance of interest. The model constructs a variety of interests, and effectively models interest information through interest alignment and interest interaction.
- We conduct extensive experiments on three real-world datasets to demonstrate the effectiveness of our model. Our model IMIN performs best when compared to other competitive networks.

2 Related Work

Traditional recommendation systems are mainly based on collaborative filtering models [19], which use the preferences of a group with similar interests and common experience to recommend information that users are interested in. It mainly includes user-based and item-based models [17]. The model based on collaborative filtering cannot solve the problem of data sparsity, so a series of models based on matrix factorization (MF) [11] appear. Matrix factorization introduces the latent factor as the implicit representation of user and item, which further improves the accuracy of model prediction. Collaborative filtering models are suitable for small-scale scenarios, but cannot be effectively applied to large-scale rich scenarios.

DNNs can effectively conduct nonlinear combination between features and model high-order complex feature interaction, which are widely used in recommendation systems [23]. A series of models fusing DNN and FM are further derived. Wide & Deep [3] learns high-order feature interactions by using multi-layer perceptron (MLP), and constructs effective feature combinations based on manual feature engineering. DeepFM [5] uses FM for the low-order combination of features on the wide side, avoiding feature engineering. DCN [20] realizes low-order and high-order combination of features by using cross layer. xDeepFM [13] introduces a compressed interaction network to generate feature interactions in an explicit fashion and at the vector-wise level.

In addition to modeling directly from the perspective of feature intersection, there are models for modeling from a matching perspective. DMF [22] adds a nonlinear MLP network to the traditional MF. NCF [7] concatenates user and item embedding vectors together, and then learns high-order interactions through MLP network. In addition, Many methods based on attention mechanism have also been proposed. AFM [21] learns the importance of different feature interactions by introducing an attention network when performing feature interactions. DIN [25] applies the attention mechanism to the user's sequential behavior, and fully extracts the user's interest information contained in the historical behavior. The introduction of interest can model user characteristics more accurately.

3 Model

In this part, we introduce the various components of the model in detail. First, we introduce the presentation layer for constructing implicit multiple interests of users and items. Then we introduce the interest alignment layer to align the interest information between users and items. Then we introduce the interest interaction and aggregation layer, which is used to effectively construct the interaction between interest features. Finally, we introduce an auxiliary loss to ensure the difference between a variety of interests.

3.1 Implicit Multiple Interest Representation Layer

User features are direct expressions of user characteristics and reflect the user's personal preferences. In actual situations, users' preferences are often diverse, which means that users have a variety of different interests. For example, the user may like fashionable and luxurious items in terms of dressing, but prefer high-quality and inexpensive items in life. In order to effectively capture the user's multiple interests, we use a multi-head self-attention mechanism to implicitly construct it. First, the self-attention mechanism based on the user's feature set is as follows:

$$A_j = softmax(\frac{Q_j^T K_j}{\sqrt{d}})V_j \tag{1}$$

where $A_j \in R^{M \times d}$ represents the feature interaction of attention perception, which can be regarded as the expression of feature information under the j-th subspace. d_k refers to the matrix dimension. Q_j, K_j, V_j are defined as follows:

$$Q_j = W_j^q Z_u, K^u = W_j^k Z_u, V^u = W_j^v Z_u \tag{2}$$

where $Z_u \in R^{M \times k}$ represents the user feature matrix. M denotes the number of user fields, and k denotes the embedding dimension of each field feature. $W_j^q, W_j^k, W_j^v \in R^{k \times d}$ represents the attention parameter matrix, and d represents the mapping dimension. Further, we use the *maxpooling* operation to extract the most important feature information from A_j^u, and use *meanpooling* to propose the averaged feature information from A_j, which is defined as follows:

$$\begin{aligned} a_j &= [a_{j,1}; a_{j,2}] \\ a_{j,1} &= max\_pooling(A_j) \\ a_{j,2} &= mean\_pooling(A_j) \end{aligned} \tag{3}$$

where $a_j \in R^{2d}$ is the user feature expression in the j-th subspace, which can be regarded as the j-th implicit interest representation of the user. Similarly, there may be multiple selected points of interest for an item. We can get $b_j \in R^{2d}$ based on the feature set of item using the above method, which represents the j-th implicit interest representation of item.

3.2 Interest Alignment Layer

The implicit multi-interest representation of user can be constructed based on user characteristics, and the implicit multi-interest representation of item can also be constructed. When a certain interest of the user matches a certain interest of the item, it means that the two have the same preference under the interest space. In order to effectively construct matching information between users' multiple interests and item multiple interests, we need to align multiple interest information. By designing user and item to perceive all the interest information of each other, information alignment can be realized, which is defined as follows:

$$\begin{aligned} a_i^{'} &= \sum_{j=1}^{l_b} \frac{exp(a_i \cdot b_j)}{\sum_{k=1}^{l_b} exp(a_i \cdot b_k)} b_j \\ b_j^{'} &= \sum_{i=1}^{l_a} \frac{exp(a_i \cdot b_j)}{\sum_{k=1}^{l_a} exp(a_k \cdot b_j)} a_i \end{aligned} \tag{4}$$

where $a_i^{'} \in R^{2d}$ represents the i-th implicit interest representation of the user who perceives the interest information of the item, and $b_i^{'} \in R^{2d}$ represents the j-th implicit interest representation of the item that perceives the user's interest information. Further, in order to avoid losing the original interest information by only using the aligned interest, the original interest information and the aligned interest information need to be fused. We use a gated network to learn the weights of different interest, which is defined as follows:

$$\begin{aligned} \gamma_i &= W_a[a_i; a_i^{'}; a_i \odot a_i^{'}; a_i - a_i^{'}] + \epsilon_a \\ a_i^f &= \gamma_i a_i + (1 - \gamma_i) a_i^{'} \end{aligned} \tag{5}$$

where $a_i^f \in R^{2d}$ represents the i-th fusion interest representation of the user, and γ_i represents the importance weight. $\odot$ stands for element-wise multiplication. The multiplication operator emphasizes the similarity of two vectors, and the subtraction operator emphasizes the difference between the two vectors. Further, we connect all the fusion interest representations of the user together, as the global interest representation of the user as follows:

$$h_u = mean\_pooling([a_1^f; a_2^f; \cdots; a_{l_a}^f]) \tag{6}$$

where $h_u \in R^{2d}$ represents the global interest representation of the user. Similarly, our fusion interest representation based on item can get the item global interest representation $h_i \in R^{2d}$.

3.3 Interest Interaction Layer

The interest representation constructed based on the implicit multi-interest representation layer only reflects the feature interaction information of the user or item itself, which does not learn the feature interaction between the user and the item. In order to effectively learn rich feature information, we construct feature interactions based on user implicit multiple interest representations and item implicit multiple interest representations, which are defined as follows:

$$S_{i,j} = a_i \odot b_j \tag{7}$$

where $S_{i,j} \in R^{2d}$ represents the interaction between the i-th interest feature of the user and the j-th interest feature of the item. Considering that the importance of different feature interactions is different, we design a compressed activation network to learn the importance of feature interactions. First, we perform the mean pooling operation on feature interaction, which is designed as follows:

$$r_{i,j} = f_{sq}(S_{i,j}) = \frac{1}{2d}\sum_{k=1}^{2d} S_{i,j}^k \tag{8}$$

Then we use the extended network to learn the importance weight, which is designed as follows:

$$A_s = f_{ex}(R) = \sigma_2(W_2\sigma_1(W_1 R)) \tag{9}$$

where $A_s \in R^{l_a \times l_b}$ represents the weight matrix, and σ_2 and σ_1 represent the activation function. We multiply the feature interaction matrix by the weight matrix, which is defined as follows:

$$S^{'} = A_s \odot S \tag{10}$$

In order to further learn the high-level interactions of features, we stitch all feature interactions together as follows:

$$s = sum\_pooling(S^{'}) \tag{11}$$

where $s \in R^{2d}$ represents the low-level feature interaction between user and item. *sum_pooling* means to sum the matrix $S^{'}$ into a one-dimensional vector. Further, we use residual network to learn low-order and high-order feature interactions, which is defined as follows:

$$x_{l+1} = f(W_l x_l + \epsilon_l) + x_l \tag{12}$$

where x_l denotes the feature representation of the l-th layer, x_0 is equal to s. The feature interaction finally learned through the L-layer network is denoted as x.

3.4 Information Aggregation Layer

The user global interest representation h_u and item global interest representation h_i are constructed through the interest alignment layer. The global interest representation represents the overall feature information of user and item. The low-order and high-order feature interaction x between user interest features and item interest features are constructed through the interest interaction layer, which reflects the feature interaction information. Taking into account the different importance of different information, we design a gated network to control the transmission of information, which is defined as follows:

$$\begin{aligned} h^{'} &= tanh(W_g x + U_g(\beta \odot h) + \epsilon_g) \\ q &= \alpha \odot h + (1-\alpha) \odot h^{'} \end{aligned} \tag{13}$$

where $q \in R^k$ represents aggregate feature information, α represents update gate, and β represents reset gate. W_g, U_g, ϵ_g represent network parameters. α and β are defined as follows:

$$\begin{aligned} \alpha &= \sigma(W_\alpha x + U_\alpha h + \epsilon_\alpha) \\ \beta &= \sigma(W_\beta x + U_\beta h + \epsilon_\beta) \end{aligned} \tag{14}$$

Based on the aggregate feature q that combines global interest information and feature interaction information, the final prediction result is:

$$p = \sigma(W_p q + \epsilon_p) \tag{15}$$

where p represents the prediction result of the model. W_p and ϵ_p represent network parameters.

3.5 Training Optimization

In this paper, we mainly predict whether the user will interact with the item, which can be regarded as a classification task. We mainly optimize logloss for classification tasks, which are defined as follows:

$$Loss_1 = -y_i log(p_i) - (1 - y_i) log(1 - p_i) \tag{16}$$

where $y_i \in \{0, 1\}$ represents the label of the sample, and p_i represents the predicted probability that the user clicks on the item. In addition, we design an implicit multi-interest representation layer for learning multi-interest representations. In order to ensure the difference of different interests, we introduce regularized auxiliary loss for multi-interest generation, which is designed as follows:

$$Loss_2 = -\lambda_1 \sum_{i=1}^{l_a} \sum_{j=i+1}^{l_a} \frac{a_i \cdot a_j}{|a_i||a_j|} - \lambda_2 \sum_{i=1}^{l_b} \sum_{j=i+1}^{l_b} \frac{b_i \cdot b_j}{|b_i||b_j|} \tag{17}$$

where a_i denotes the i-th implicit interest representation of the user, and b_j represents the j-th implicit interest representation of item. λ_1 and λ_2 represent hyperparameters, which are used to control the degree of regularization. Finally, we fuse log loss and auxiliary loss to get the total loss as follows:

$$Loss = \frac{1}{N} \sum_{i=1}^{N} Loss_1 + Loss_2 \tag{18}$$

where $Loss$ represents the total loss, and N denotes the number of samples.

4 Experiment

In this section, we solve the following problems by designing different experiments:

- **Q1** How does our proposed IMIN compare to state-of-the-art models?
- **Q2** Do the various modules and strategies we propose really make sense to improve the effect of the model?

4.1 Experimental Settings

Datasets. We evaluated model performance on the following data sets:

Frappe[1] Dataset [1]. Frappe is a context-aware mobile app recommender system. Frappe dataset are composed of application logs that contain ID information, weather information, and other rich contextual information. The dataset contains 192,406 samples.

[1] https://www.baltrunas.info/research-menu/frappe.

MovieLens[2] Dataset [6]. It is the baseline dataset in the recommendation scenario. MovieLens contains a lot of movie recommendation data, including user information, movie information, time information and other rich features. We use the dataset of one million samples.

Criteo[3] Dataset. It is an open industry benchmark dataset used to develop models for predicting ad click-through rates, which contains 45million samples. The dataset describes the prediction of the probability of clicking on the advertisement on the page given a user and the visited page.

Evaluation Metrics. We use AUC to measure model performance. In practical scenarios, positive and negative samples are often unbalanced, and AUC can evaluate model performance well in this case. In addition, in order to visually show the improvement degree of different models compared with the benchmark model, we introduce RelaImpr [8] metric, which is defined as follows:

$$RelaImpr = (\frac{AUC(measured\_model) - 0.5}{AUC(base\_model) - 0.5} - 1) \times 100\% \tag{19}$$

Baselines. The competitive models we compare are as follows:

– FM [15]: FM uses latent vector representation to carry out the second-order interaction between features, which fully improves the model's predictive ability. FM is a widely used benchmark model.
– DNN [4]: DNN has powerful representation and fitting capabilities, and can fully learn the implicit interactions between features. It is a benchmark model based on neural networks.
– Wide & Deep [3]: It includes both the explicit feature interaction constructed by feature engineering and the implicit feature interaction constructed by neural network. It can significantly enhance the model learning ability.
– NFM [7]: NFM first performs second-order feature interaction at the vector level, and further learns complex interactions through neural networks.
– DeepFM [5]: DeepFM combines the second-order explicit feature interaction constructed by FM and the implicit feature interaction constructed by DNN.
– DCN [20]: DCN has designed a network that explicitly constructs low-order and high-order feature interactions, which can effectively construct feature interaction information.
– xDeepFM [13]: xDeepFM introduces a compressed interaction network to generate feature interactions in an explicit fashion and at the vector-wise level. xDeepFM can learn low-order and high-order feature interactions explicitly and implicitly.
– FiBiNET [9]: FiBiNET can dynamically learn the importance of features via the Squeeze-Excitation network mechanism. It is able to effectively learn the feature interactions via bilinear function.

[2] https://grouplens.org/datasets/movielens/.
[3] http://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/.

Parameter Settings. We take 90% of each dataset as the training set, 10% as the validation set, and 10% as the test set. The validation set is mainly used for hyperparameter selection. The embedding size is selected in [8, 16, 32, 64, 128, 256, 512]. The number of hidden layers is adjusted sequentially from 1 to 5. For model optimization, we uniformly adopt Adam [10], which is a widely used optimizer. Considering the running time and efficiency, we set the batch size to 512, and select the learning rate in [0.0005, 0.001, 0.005, 0.01, 0.05, 0.1]. Neural network models all use dropout and batch normalization strategies to prevent overfitting. The dropout rate is adjusted sequentially from 0 to 0.5. In addition, we also use an early stop strategy during the training process, and stop training when the model does not improve in 5 consecutive epochs. For simplicity, we set the embedding size and hidden layer size to be the same. For IMIN, we set the number of implicit interests of users and items to be the same, and keep the hidden layer and embedding dimensions consistent.

4.2 Performance Comparison (Q1)

To verify the effectiveness of our model, we conducted comprehensive experiments on three real datasets. The experimental results are shown in Table 1. First, we can see that compared to the second-order feature interaction model FM and the implicit high-order feature interaction model DNN, models that integrate low-order and high-order feature interactions, such as xDeepFM and DCN, perform better. Second, our model surprisingly outperforms other state-of-the-art models on all three datasets. The AUC of our IMIN on the Frappe dataset is 0.9891, which is significantly higher than that of xDeepFM (0.9858), and is 3.03% higher than that of FM on RelaImpr. Since our model takes into account various interests of users and items, the matching relationship between interests is fully established. The experimental results fully verify the effectiveness of our model.

Table 1. Performance of different models on MovieLens, Frappe and Criteo datasets.

Models	MovieLens		Frappe		Criteo	
	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr
FM	0.8072	0.000%	0.9795	0.00%	0.7924	0.00%
DNN	0.8082	0.33%	0.9814	0.40%	0.8004	2.74%
Wide& Deep	0.8105	1.07%	0.9838	0.90%	0.8017	3.18%
PNN	0.8113	1.34%	0.9845	1.04%	0.8028	3.56%
NFM	0.8116	1.43%	0.9844	1.02%	0.8031	3.66%
DeepFM	0.8121	1.60%	0.9847	1.08%	0.8035	3.80%
DCN	0.8118	1.49%	0.9849	1.13%	0.8042	4.04%
xDeepFM	0.8129	1.86%	0.9858	1.31%	0.8069	4.96%
FiBiNET	0.8144	2.34%	0.9872	1.61%	0.8081	5.37%
IMIN	**0.8165**	**3.03%**	**0.9891**	**2.00%**	**0.8095**	**5.85%**

4.3 Model Ablation Analysis (Q2)

In order to study whether each module is really meaningful for the improvement of the model effect, we conducted a comprehensive ablation experiment. First, we remove interest alignment layer (IAL) to explore the role of interest alignment between users and items, and then we remove the interest interaction layer (IIL) to explore the role of interest feature interactions. Second, we replaced the gating fusion mechanism with vector addition and vector connection respectively to explore the effect of the gating mechanism. We conducted experiments on the three datasets, and some of the results are shown in Fig. 1.

- we can see that the effect of the model is worse after removing interest alignment and interaction. Although the effect of interest interaction layer is better than interest alignment, interest alignment layer can be used as supplementary information to further enhance the model effect.
- Whether it is vector addition or vector connection, the performance of the model is worse. This shows that there is a gap in the feature interaction of different perspectives and cannot be integrated in a simple and direct way. The results further verify the effectiveness of our gating mechanism.

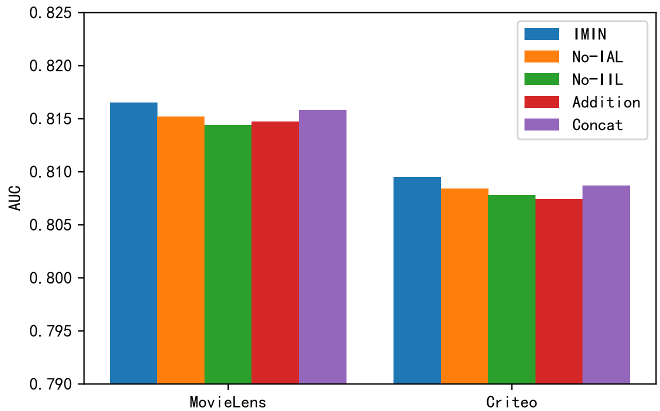

Fig. 1. Performance of different modules on MovieLens and Criteo datasets.

5 Conclusion

In this paper, we propose an implicit multiple interest network (IMIN), taking into account the importance of interest. The model constructs a variety of interests, and effectively models interest information through interest alignment and interest interaction. We conduct extensive experiments on three real-world datasets to demonstrate the effectiveness of our model. Our model IMIN performs best when compared to other competitive networks.

There are two directions for future study. First, we consider introducing time series information to enrich interest modeling. We can expand the user's interest into long-term and short-term dynamic interest, thereby further improving the accuracy of prediction. Second, we consider the introduction of multi-modal features. By introducing modal information such as pictures, texts, etc., item features can be modeled more accurately.

Acknowledgments. This work is supported by the National Key Research and Development Program of China under Grant No. 2019YFF0301801, and the National Natural Science Foundation of China under Grant No. 61806199.

References

1. Baltrunas, L., Church, K., Karatzoglou, A., Oliver, N.: Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. arXiv preprint arXiv:1505.03014 (2015)
2. Bobadilla, J., Ortega, F., Hernando, A., Gutierrez, A.: Recommender systems survey. Knowl.-based Syst. **46**, 109–132 (2013)
3. Cheng, H.T., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10 (2016)
4. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference On Recommender Systems, pp. 191–198 (2016)
5. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247 (2017)
6. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. (tiis) **5**(4), 1–19 (2015)
7. 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)
8. He, X., et al.: Practical lessons from predicting clicks on ads at facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, pp. 1–9 (2014)
9. Huang, T., Zhang, Z., Zhang, J.: Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 169–177 (2019)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
12. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
13. Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1754–1763 (2018)

14. Qu, Y., et al.: Product-based neural networks for user response prediction. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1149–1154. IEEE (2016)
15. Rendle, S.: Factorization machines. In: 2010 IEEE International Conference on Data Mining, pp. 995–1000. IEEE (2010)
16. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_1
17. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295 (2001)
18. Sarwar, B.M.: Sparsity, scalability, and distribution in recommender systems (2001)
19. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence 2009 (2009)
20. Wang, R., Fu, B., Fu, G., Wang, M.: Deep & cross network for ad click predictions. In: Proceedings of the ADKDD'17, pp. 1–7 (2017)
21. Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., Chua, T.S.: Attentional factorization machines: Learning the weight of feature interactions via attention networks. arXiv preprint arXiv:1708.04617 (2017)
22. Xue, H.J., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: IJCAI, vol. 17, pp. 3203–3209. Melbourne, Australia (2017)
23. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. (CSUR) **52**(1), 1–38 (2019)
24. Zhang, W., Du, T., Wang, J.: Deep learning over multi-field categorical Data. In: Ferro, N., et al. (eds.) ECIR 2016. LNCS, vol. 9626, pp. 45–57. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30671-1_4
25. Zhou, G., et al.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1059–1068 (2018)