Leveraging Multiple Implicit Feedback for Personalized Recommendation with Neural Network

Hongfa Wen
School of Automation
Hangzhou Dianzi University
Hangzhou, P.R. China
hf_wen@outlook.com

Xin Liu*
School of Automation
Hangzhou Dianzi University
Hangzhou, P.R. China
xinliu@hdu.edu.cn

Chenggang Yan
School of Automation
Hangzhou Dianzi University
Hangzhou, P.R. China
cgyan@hdu.edu.cn

Linhua Jiang
Information Science and Technology Research
Shanghai Advanced Research Institute, Chinese Academy of Sciences
Shanghai, P.R. China
jianglh@sari.ac.cn

Yaoqi Sun
School of Automation
Hangzhou Dianzi University
Hangzhou, P.R. China
syq@hdu.edu.cn

Jiyong Zhang
School of Automation
Hangzhou Dianzi University
Hangzhou, P.R. China
jzhang@hdu.edu.cn

Haibing Yin
School of Communication Engineering
Hangzhou Dianzi University
Hangzhou, P.R. China
yhb@hdu.edu.cn

Abstract

In recent years, deep neural networks have been widely applied on recommender systems. Most research efforts are put on modeling the side information such as textual information, contextual information and social network information, but the core part, i.e., interaction relationship between users and items are relatively less explored by neural networks, in particular, when multiple types of implicit feedbacks, e.g., click, browsing, add-to-cart, etc. are available in the system. In this paper, we propose an end-to-end learning framework, which systematically and comprehensively models multiple implicit feedback between users and items. Firstly, for each type of implicit feedback, we apply matrix factorization and Multi-Layer Perception (MLP) to capture both linearity and nonlinearity of user-item interactions. Then we fuse the effects of multiple implicit feedback through neural networks to boost the quality of recommendation. Experiments on Alibaba real production dataset with over two million interactions demonstrate the effectiveness of proposed approaches, which achieve superior performance compared with state-of-the-art methods.

AIAM 2019, October 17–19, 2019, Dublin, Ireland

https://doi.org/10.1145/3358331.3358337

CCS CONCEPTS

Information systems → Information retrieval → Retrieval tasks and goals → Recommender systems

KEYWORDS

Multi-Layer perception; Matrix factorization; Neural networks; Multiple implicit feedback

ACM Reference format:

Hongfa Wen, Xin Liu, Chenggang Yan, Linhua Jiang, Yaoqi Sun, Jiyong Zhang and Haibing Yin. 2019. Leveraging Multiple Implicit Feedback for Personalized Recommendation with Neural Network. In *AIAM' 19: 2019 International Conference on Artificial Intelligence and Advanced Manufacturing, October 17–19, 2019, Dublin, Ireland. ACM, New York, NY, USA, 6 pages.* https://doi.org/10.1145/3358331.3358337

1 INTRODUCTION

By suggesting personalized information to individual users, recommender systems (RSs) have become an effective tool to handle information overload in many online application scenarios

such as product recommendation at Amazon [2], video recommendation at YouTube [33], movie recommendation at Netflix [3, 4], to name a few. Since a user's preference can be inferred according to her behaviors along the history, exploiting historical behavior data, e.g., by collaborative filtering has become the fundamental way to generate recommendations [1, 4, 10].

Among various collaborative filtering techniques, matrix factorization (MF) was one of the mainstream methods thanks to the well-known 1 million US dollar competition prize offered by Netflix [31]. The basic idea of MF is to project users and items into a shared space, and model a user's preference on an item by inner product of the corresponding latent factor vectors. MF brings in the properties of simplicity, effectiveness and scalability for the recommendation problem, but its performance is limited by the linear way of user-item interaction modeling, i.e., inner product. With the successful applications of deep learning in the area of Computer Vision [9, 11, 13, 23], Natural Language Processing (NLP) [5, 27], etc., incorporating deep neural network component into recommendation models to capture complex structure of interaction data has become a trend in the community, and evident improvements have been observed [7, 12]. For instance, Google's deep & wide model [6] integrates generalized linear model and multilayer perception to model the interactions among contextual information, DeepFM [7], xDeepFM [8], etc., merges Factorization Machines (FM) [29] and neural networks to improves the modeling capacity of original FM.

However, existing collaborative filtering models suffer from the following limitations: (1) most deep learning based methods focus on modeling users' behavior data and various side information such as temporal information, texts, social network information, contextual information, etc., but the sophisticated modeling of the pure user-item interaction data is relatively less explored; (2) most collaborative filtering models, including both MF based and deep learning based ones, typically focus on one type of behavior, or rely on simple linear combination of multiple types of behavior, lacking the effective way to capture the complex relationships among different types of behavior, e.g., how click, browse and add-to-cart behavior altogether reflect and impact users' purchase behavior?

In this work, we focus on implicit feedback, e.g., clicking an article, which is more common in many application scenarios compared to explicit feedback like ratings, and propose an end-to-end learning model to systematically and comprehensively models multiple implicit feedback between users and items to improve recommendation quality. Firstly, for each type of implicit feedback, we merge MF and Multi-Layer Perception (MLP) to capture the strengths of MF's linearity and MLP's nonlinearity to comprehensively model the complex user-item interactions. Then we fuse the effects of multiple implicit feedback through neural networks to further boost the accuracy of recommendations. We conduct extensive experiments over large-scale real datasets and demonstrate the effectiveness of our proposed models by comparing with the state-of-the-art collaborative filtering models for top-N recommendation task.

2 RELATED WORKS

In recommender systems, there exist two types of user behavior feedback, i.e., explicit feedback [14, 15, 30, 35], like rating, reviews and implicit feedback [16, 17, 18], like clicking, browsing, etc. Early works focus on explicit feedback, while recent attention has shifted to implicit data, which is more pervasive in systems. For instance, Liu et al. [16] proposed a boosting algorithm which uses a re-weighting strategy to build multiple component recommenders that assigns a dynamic weight distribution to observed user-item implicit feedback. Yu et al. [17] systematically extracted latent features and used Bayesian ranking optimization techniques to build the model. He et al. [32] designed an element-wise Alternating Least Squares (eALS) based algorithm to effectively optimizes MF model with variable weighted missing data from implicit feedback.

In recent years, with the fast development of deep learning techniques in various fields such as Computer Vision, applying deep neural networks to build recommendation models brings new opportunities to the community. For instance, He et al. [12] designed a model that uses neural networks instead of inner products to learn the complex interactions between users and items. Guo et al. [7] proposed a new neural network architecture that models low-order and high-order feature interactions by FM and Deep Neural Networks (DNNs), respectively. Yi et al. [37] built two feature transforming functions to generate latent factors between users and items directly from various input information, and converted high-dimensional and sparse implicit feedback information into low-dimensional real-value vectors that retain the main features through the Implicit Feedback Embedding (IFE) module.

Most existing works typically focus on one type of feedback. However, it is common that multiple types of feedback exist in systems, ignoring which will cause information loss and degrade the quality of recommendations. A few works attempt to combine multiple types of feedback, for instance, Chen et al. [24] proposed a probabilistic MF based latent factor model that decomposes both explicit and implicit feedback matrices into a shared subspace. Liu et al. [19] studied the personalized ranking recommendation problem by integrating one type of explicit feedback and multiple types of implicit feedback. Gao et al. [36] modeled complex multi-type interactions between users and items by considering cascading relationships between different types of behaviors. However, existing models typically rely on simple linear methods to combine feedback. In the next section, we describe our approach, which leverages deep neural networks and matrix factorization to comprehensively capture the complex relationships among different types of feedback for personalized recommendation.

3 OUR APPROACH

In this section, we first discuss the multiple implicit feedback problem for personalized top-N recommendation. Then, we show how to use MF to capture linear structure of user-item interactions. Next, we propose a multi-branch MLP based model to learn nonlinear functions between users and items, and also across multiple types of users' behavior. Finally, we present an ensemble network architecture that unifies MF component and MLP component to further enhance user-item interaction modeling capacity.

3.1 Multiple Implicit Feedback for Top-N Recommendation

The pervasively available implicit feedback data has become an important information source for building personalized top-N recommender systems. However, sophisticatedly modeling multiple types of implicit feedback simultaneously is relatively less explored and has recently started receiving attention in the community [19]. In this work, in the presence of multiple types of implicit feedback, we use one of them as our optimization and recommendation objective, and the remaining as auxiliary information to support the recommendation objective. The former feedback is defined as target feedback and the latter feedback is defined as supportive feedback.

For instance, in product recommendation, we use purchase behavior as target feedback, where our goal is to predict users' next purchase behavior; other feedback behaviors such as click, favorite, add-to-cart, etc. are used as supportive feedback to improve the model prediction accuracy.

In order to provide top-N recommendation ranking, we follow the pairwise learning strategy where the optimal ordering of a pair of items are learned for ranking. It is therefore important to generate high quality positive-negative item pairs for model training. Given a positive item, i.e., from target feedback, we propose a sampling approach where the items that the user only interacted through supportive feedback are sampled as negative items. Compared to traditional random sampling [20], such a method can more accurately select informative negative items, which has been verified by experiments (see Section IV).

3.2 Matrix Factorization (MF)

MF has been extensively and effectively applied in recommender systems. Intuitively, by mapping users and items to a shared latent space with dimensionality of d, MF models a user's preference for an item as the inner product of the corresponding latent factor vectors in that space. Specifically, each user u and item i are represented by vectors $p_u \in R^d$ and $q_i \in R^d$, respectively. For a given user u, the elements of p_u measure the user's interest level in the corresponding factors, and q_i expresses item i's characteristics from different aspects. The resulting dot product y_{ui} expresses user u's preference for item i, calculated as:

$$\hat{y}_{ui}=q_i^T p_u=\sum_{t=1}^{d} q_{it}\, p_{ut} \tag{1}$$

As we can see, MF can be regarded as a linear model of latent factors of users and items.

3.3 Multi-branch Multi-Layer Perceptron (M-MLP)

MF model effectively captures the linearity of user-item interaction relationships. However, in practice, user-item interactions typically reflect complex nonlinear structure; moreover, modeling the mutual influence among different types of feedback also requires sophisticated modeling beyond linearity. To address these issues, we propose to use MLP to learn the interaction functions between latent features of users and items as well as the mutual influence among different types of implicit feedback.

As shown in Fig. 1, our model has a multi-branch network structure, named Multi-branch Multi-Layer Perceptron (M-MLP), in which each branch corresponds to a type of implicit feedback behavior and learns complex interaction structure between users and items under this implicit feedback. Specifically, we use "Target" to symbolize the target feedback, and abbreviate each type of supportive implicit feedback behavior into typeA, typeB, and typeC, respectively. Note that for the purpose of presentation, we use three types of supportive feedback, but in practice, any number of supportive feedback types can be supported, subject to the complexity of the model.

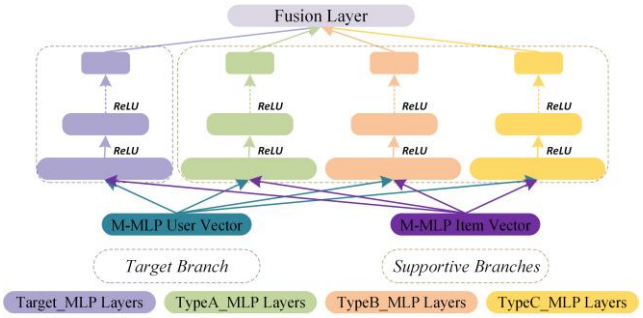

Figure 1: Multi-branch Multi-Layer Perceptron Model. Best viewed in color

Finally, we merge the output of multiple branch networks by a fully connected layer, which generates the output of the whole M-MLP model. In consideration of the sparsity of our data, and in order to prevent the model from overfitting, we use ReLU as the activation function in each layer. More precisely, our M-MLP model is defined as follows, where (2) and (3) show the forward propagation in each branch from the second layer to the L-th layer. Equation (4) shows the fusion of branches representing different implicit feedbacks,

$$y_2^t=\sigma_2^t\left(w_2^{t\,T} y_1+b_2^t\right), \tag{2}$$

.....

$$y_L^t=\sigma_L^t\left(w_L^{t\,T} y_{L-1}^t+b_L^t\right), \tag{3}$$

$$\hat{y}_{ui}=\sigma\left(w^T \mathrm{concat}\left(y_L^{\{t\}}\right)+b\right), \tag{4}$$

where $y_1=\left[p_u^{MLP}, q_i^{MLP}\right]^T$, L represents the number of layers of each MLP branch network, and w_x^t, b_x^t, and σ_x^t denote the weight matrix, bias vector, and activation function for the x-th layer of each branch perceptron, respectively. Particularly, we define $t \in \{\text{Target, typeA, typeB, typeC}\}$, and $y_L^{\{t\}}=\left(y_L^{\{\text{Target}\}}, y_L^{\{\text{typeA}\}}, y_L^{\{\text{typeB}\}}, y_L^{\{\text{typeC}\}}\right)$, and concat represents the concatenation operation of different branch networks. It should be noted that during the training phase, the parameter updates of different branch networks are asynchronous. For the target feedback, the parameter update is performed on the interaction data of each user and item. For the supportive feedback branches, the parameters are updated only when there is feedback for this type behavior between users and items, otherwise they are not updated.

3.4 Fusing MF and M-MLP

In previous subsections, we have discussed two different models MF and M-MLP to learn the interaction relationships between users and items, capturing both linearity and nonlinearity aspects. So far, MF and M-MLP are still two independent components, so how to make full use of the strengthen of the two is of particularly critical.

In order to provide enhanced modeling capacity, we propose an ensemble network architecture to fuse MF and M-MLP. Specifically, we assign a latent factor vector to each user and item for MF component and M-MLP component respectively. That is, each user and item has a latent representation for MF and a latent representation for M-MLP (see Fig. 2). Equation (5) shows that the output of MF component is obtained by conducting the element-wise product of the corresponding latent factor vectors:

$$\hat{y}_{MF}=p_u^{MF}\cdot q_i^{MF}\ , \tag{5}$$

where p_u^{MF} and q_i^{MF} denote the user and item latent factor vectors for MF part. Similarly, the output of M-MLP component is calculated as follows:

$$\hat{y}_{M\text{-}MLP} = \sigma_M\left(w_M^T\left(\text{concat}\left(\sigma_L^{\{t\}}\left(w_L^{\{t\}^T}\left(\ldots\sigma_2^{\{t\}}\left(w_2^{\{t\}^T}\begin{bmatrix}p_u^{MLP}\\ q_i^{MLP}\end{bmatrix}+b_2^{\{t\}}\right)\ldots\right)+b_L^{\{t\}}\right)\right)\right)+b_M\right), \tag{6}$$

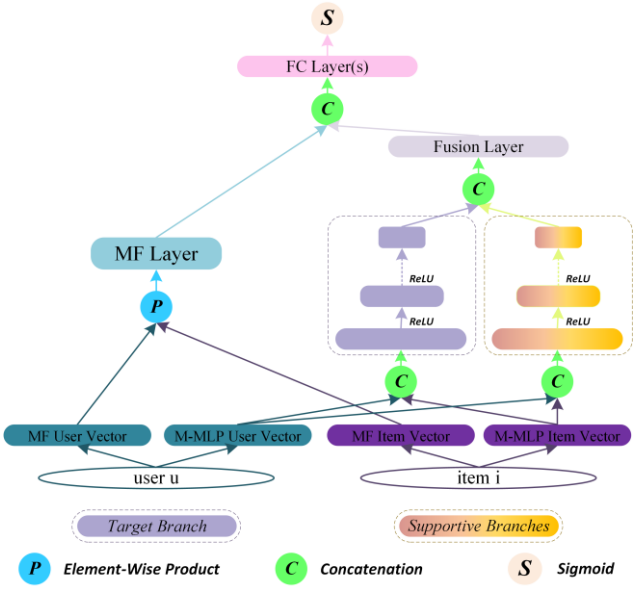

Figure 2: Our ensemble network architecture. Gradient color represents multiple supportive branches. Best viewed in color

where p_u^{MLP} and q_i^{MLP} denote the user and item latent factor vectors for M-MLP part. Note that the dimensionality of the latent factor vector of MF and M-MLP may not share the same size, which is more flexible to characterize different latent aspects of the two different models. Finally, MF component and M-MLP component are fused by concatenating the corresponding output, which is then feed into another MLP (a fully connected layer) to combine the linearity of MF and nonlinearity of M-MLP to model complex user-item interaction relationships. Equation (7) shows how the two components are fused by a fully connected layer.

$$\hat{y}_{ui}=\sigma\left(w^T\begin{bmatrix}\hat{y}_{GMF}\\ \hat{y}_{M\text{-}MLP}\end{bmatrix}\right) \tag{7}$$

The entire model can be optimized with the standard neural network back-propagation algorithm. We find Adam algorithm [26] produces the best results, which will be demonstrated in the next section.

4 EXPERIMENTS

In this section, we conduct extensive experiments over a real dataset to demonstrate the effectiveness of our proposed architecture.

4.1 Experimental Settings

We use Taobao user behavior data [21, 22], where we randomly select about 20 thousand users. Four types of implicit feedback are extracted from the data, i.e., clicking, purchasing, adding item to shopping cart and item favoring. Each record of the data represents a user-item interaction, consisting of user ID, item ID, feedback type and timestamp. TABLE 1 summarizes the statistics of the dataset after preprocessing.

We compare the proposed approach with the following representative methods:

Table 1: Statistics of the dataset after preprocessing

UserBehavior dataset			
Users#	Item#	Feedback type#	Records#
19576	629758	4	2003670

1) BPR: This model [25] optimizes MF model with a pairwise ranking loss, which is tailored to learn from implicit feedback.

2) MFPR: Based on BPR, this model [19] integrates explicit feedback and multiple implicit feedback. Since rating information is not available in the dataset, we only adapt implicit feedback modeling part of this model in our comparison study.

3) WMF: This model [7] treats the feedback as the indication of positive and negative preference associated with vastly varying confidence levels.

4) NCF: This model [12] replaces the inner product with a neural architecture that can learn an arbitrary function from data. Similar to our approach, both MF and Multi-layer perceptron are leveraged to learn the user-item interaction functions, but the network is designed to handle a single type of implicit feedback.

In the experiments, every model focuses on predicting a user's next purchase behavior. BPR, WMF and NCF only rely on historical purchase data to build model, while our approach and MFPR leverages all four types of feedback.

In order to compare the performance of the models, we adopt the widely-used leave-one-out evaluation strategy [26, 32, 34]. For each user, we extract her latest interaction for testing, and use the remaining interaction records as the training set. Since it is too time-consuming and impractical to rank all items for every user during evaluation, we followed the common strategy [12] that randomly samples 99 items that are not interacted by the users. We combined the 99 items and test item to get the final predicted rankings. We use Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [28] to evaluate the performance of the models. Specifically, HR@10 intuitively measures whether the test item is present on the top-10 recommendations, and the NDCG@10 measures the ranking quality which assigns higher scores to hits at top position ranks. We calculate all above two metrics for each test user and report their average scores.

Our approach is implemented using PyTorch (https://pytorch.org/). For model initialization, we randomly initialized neural network parameters like weights and biases with a Gaussian distribution

(with a mean of 0 and standard deviation of 0.01). The model is optimized with mini-batch Adam [26]. We set the batch size to 1, and set the learning rate to 0.001. All experiments were conducted on a NVIDIA TITAN X Pascal GPU.

4.2 Design Validation

We first demonstrate how the accuracy of our approach varies with different dimensionality of the latent factor vector. For simplicity, we assume MF component and M-MLP component share the same embedding size. As summarized in TABLE 2, our model achieves better performance when the number of factors increases from 2 to 4, indicating more factors improves the modeling capacity. The performance starts slightly decreasing when the factor number becomes further larger. This is because the network becomes over-complicated with more factors, causing the overfitting issue.

Table 2: Results for our model with different factors

Factors	2	4	8	16	32
HR@10	0.2739	0.2759	0.2695	0.2623	0.2546
NDCG@10	0.1638	0.1664	0.1646	0.1631	0.1566

We next apply different negative sampling ratios to study how the negative sampling strategy influences the recommendation performance. The results are summarized in TABLE 3, where ng-x indicate x negative instances are sampled for each positive instance. We observe that higher negative sampling ratio boosts the performance, but too many negative instances may introduce sample noises and decreases the accuracy. In our experiments, our model achieves the best performance when the number of negative samples is set to 3.

In our model, the MLP component models user-item interactions through neural network with hidden layers. We finally investigate how the number of hidden layers impacts our model. Fig. 3 shows when different number of layers is applied, how the performance of our model evolves with different training iterations. We observe from the figure that stacking more layers are beneficial to performance, and our model with 2-layers achieves the best performance. This result indicating the effectiveness of using deep learning based models for collaborative filtering. Again, controlling neural network complexity, i.e., the number of layers is important to avoid overfitting issues.

Table 3: Results for our model with different negative sampling ratios

HR@10			
Factors	4	8	16
ng-1	0.2694	0.24	0.2291
ng-2	0.2735	0.2643	0.242
ng-3	0.276	0.2733	0.2565
ng-4	0.2759	0.2728	0.2543

NDCG@10			
Factors	4	8	16
ng-1	0.1665	0.1534	0.1499
ng-2	0.1663	0.1634	0.1523
ng-3	0.1654	0.1639	0.159
ng-4	0.1664	0.163	0.1558

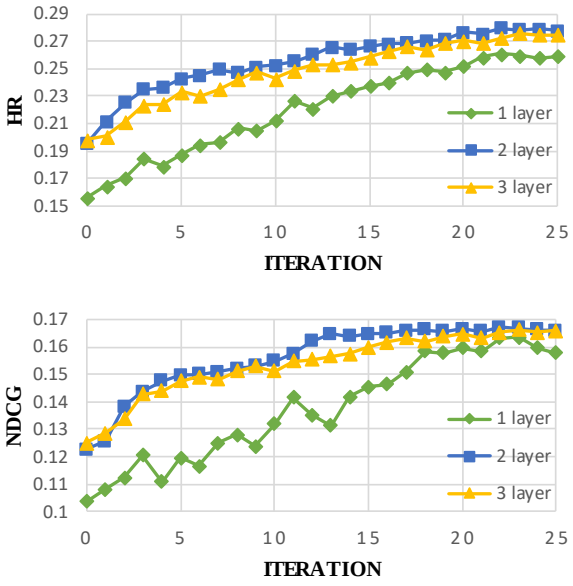

Figure 3: Performance of our model w.r.t. the number of layers in neural network

4.3 Performance Comparison

In this subsection, we compare the proposed approach with the representative models. For our approach, optimal parameters are configured as demonstrated in previous subsection. Parameters of other models such as latent factor vector dimensionality are set following original papers; if certain parameters are not mentioned in original papers, they are tuned by cross-validation. Noted that we run each method for 10 times and report the average scores; standard error is also calculated (range from 0.0050 to 0.0085) to demonstrate the comparison is statistically significant.

TABLE 4 shows the performance comparison in terms of HR@10 and NDCG@10. The baseline models BPR and WMF performs similarly. To be specific, WMF is slightly better in terms of HR@10, while BRP outperforms WMF when NDCG is used, indicating BPR's pairwise ranking loss is more beneficial to ranking task. By integrating multiple types of implicit feedback, MFPR consistently outperforms BPR and WMF, proving the importance of using multiple implicit feedback. On the other hand, deep learning based models NCF and our approach significantly outperforms the traditional methods that rely on MF model only, demonstrating that deep neural network is capable of capturing more complex latent aspects of user-item interaction structure. Although both NCF and our approach utilize MF and deep neural networks to model user-item interactions, NCF relies on only one type of implicit feedback while our approach makes use of all four types of feedback. Therefore, experimental results show that our approach outperforms NCF in terms of both HR@10 and NDCG@10. In particularly, when HR@10 is used, our approach improves NCF by 11.8%.

Table 4: Results of hr@10 and ndcg@10 with different methods.

Methods	MFPR	BPR	WMF	NCF	Our approach

HR@10	0.2426	0.2296	0.2315	0.2500	0.2795
NDCG@10	0.1568	0.1505	0.1485	0.1670	0.1673

5 CONCLUSION

In this work, we propose an end-to-end learning framework to integrate multiple implicit feedback to boost top-N personalized recommendation. MF and MLP are leveraged to capture the linearity and non-linearity of the complex user-item interaction relationships respectively. An ensemble architecture is designed to merge MF component and MLP component to further improve modeling capacity. Extensive experiments on real world dataset confirm the superiority of our approach. As for future directions, we intend to investigate more advanced deep neural network techniques such as attention mechanisms to improve recommendation performance.

ACKNOWLEDGMENT

We would like to thank Xiaofei Zhou and Jia Hou for participating in discussions and for providing helpful comments on the paper draft.

This work is supported, in part, by the National Key Research and Development Program of China under Grant 2017YFC0820605.

This work is also partially supported by Zhejiang Province Nature Science Foundation of China LR17F030006, National Nature Science Foundation of China (61671196, 61525206, 61701149), National Key Research and Development Program of China (2017YFC0820600), 111 Project, No. D17019.

This work is also supported, in part, by the National Natural Science Major Foundation of Research Instrumentation of P.R. China under Grants 61427808.

REFERENCES

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," In Proceedings of the 10th WWW, 2001.

[2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," IEEE Internet Computing, vol. 7, pp. 76-80, February 2003.

[3] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," In Proceedings of the 24th ICML, 2007.

[4] Y. Koren, "Collaborative filtering with temporal dynamics," In Proceedings of the 15th ACM SIGKDD, 2009.

[5] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," In Proceedings of the 25th ICML, 2008.

[6] H.T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, and H. Aradhye, et al., "Wide & deep learning for recommender systems," In Proceedings of the 1st DLRS, 2016.

[7] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," In Proceedings of the 26th IJCAI, 2017.

[8] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "Xdeepfm: combining explicit and implicit feature interactions for recommender systems," In Proceedings of the 24th ACM SIGKDD, 2018.

[9] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," IEEE Transactions on Intelligent Transportation Systems, 2018.

[10] X. Liu and W. Wu, "Learning Context-aware Latent Representations for Context-aware Collaborative Filtering," In Proceedings of the 38th ACM SIGIR, 2015.

[11] C. Yan, H. Xie, S. Liu, J. Yin, Y. Zhang, and Q. Dai, "Effective uyghur language text detection in complex background images for traffic prompt identification," IEEE Transactions on Intelligent Transportation Systems, 2018.

[12] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.S. Chua, "Neural collaborative filtering," In Proceedings of the 26th WWW, 2017.

[13] C. Yan, H. Xie, J. Chen, Z. Zha, X. Hao, and Y. Zhang, et al., "A fast uyghur text detector for complex background images," IEEE Transactions on Multimedia, 2018.

[14] L. Wu, E. Chen, Q. Liu, L. Xu, T. Bao, and L. Zhang, "Leveraging tagging for neighborhood-aware probabilistic matrix factorization," In Proceedings of the 21st ACM CIKM, 2012.

[15] Y. Zhen, W.J. Li, and D.Y. Yeung, "TagiCoFi: tag informed collaborative filtering," In Proceedings of the 3rd ACM RecSys, 2009.

[16] R. Yu, Y. Zhang, Y. Ye, L. Wu, C. Wang, and Q. Liu, et al., "A boosting algorithm for item recommendation with implicit feedback," In Proceedings of the 27th ACM CIKM, 2018.

[17] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, and Q. Gu, et al., "Recommendation in heterogeneous information networks with implicit user feedback," In Proceedings of the 7th ACM RecSys, 2013.

[18] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2009.

[19] J. Liu, C. Shi, B. Hu, S. Liu and P.S. Yu, "Personalized ranking recommendation via integrating multiple feedbacks," In Advances in Knowledge Discovery and Data Mining, 2017.

[20] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "TFMAP: optimizing MAP for top-n context-aware recommendation," In Proceedings of the 35th ACM SIGIR, 2012.

[21] H. Zhu, X. Li, P. Zhang, G. Li, J. He, and H. Li, et al., "Learning tree-based deep model for recommender systems," In Proceedings of the 24th ACM SIGKDD, 2018.

[22] H. Zhu, D. Chang, Z. Xu, P. Zhang, X. Li, and J. He, et al., "Joint optimization of tree-based index and deep model for recommender systems," 2019.

[23] C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang, and Q. Dai. "Cross-modality bridging and knowledge transferring for image understanding," IEEE Transactions on Multimedia, 2018.

[24] S. Chen and Y. Peng, "Matrix factorization for recommendation with explicit and implicit feedback," Knowledge-Based Systems, 2018.

[25] S. Rendle, C. Freudenthaler, Z. Gantner, and L.S. Thieme, "BPR: bayesian personalized ranking from implicit feedback," In Proceedings of the 25th UAI, 2009.

[26] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," In Proceedings of the 3rd ICLR, 2015.

[27] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, and I. Gulrajani, et al., "Ask me anything: dynamic memory networks for natural language processing," In Proceedings of the 33rd ICML, 2016.

[28] X. He, T. Chen, M.Y. Kan, and X. Chen, "TriRank: review-aware explainable recommendation by modeling aspects," In Proceedings of the 24th ACM CIKM, 2015.

[29] S. Rendle, "Factorization machines with libFM," ACM Transactions on Intelligent Systems and Technology, vol. 3, pp. 1-22, 2012.

[30] H. Ma, D. Zhou, C. Liu, M.R. Lyu, and I. King, "Recommender systems with social regularization," In Proceedings of the 4th ACM WSDM, 2011.

[31] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, IEEE, vol. 42, pp. 30–37, 2009.

[32] X. He, H. Zhang, M.Y. Kan, and T.S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," In Proceedings of the 39th ACM SIGIR, 2016.

[33] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," In Proceedings of the 10th RecSys, 2016.

[34] I. Bayer, X. He, B. Kanagal, and S. Rendle, "A generic coordinate descent framework for learning from implicit feedback," In Proceedings of the 26th WWW, 2017.

[35] X. Liu and K. Aberer, "Towards a dynamic top-N recommendation framework," In Proceedings of the 8th ACM RecSys, 2014.

[36] C. Gao, X. He, D. Gan, X. Chen, F. Feng and Y. Li, et al., "Learning recommender systems from multi-behavior data," In Proceedings of the 35th ICDE, 2019.

[37] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang and S. Liu, et al., "Deep matrix factorization with implicit feedback embedding for recommendation system," IEEE Transactions on Industrial Informatics, 2019.