2019 IEEE 35th International Conference on Data Engineering (ICDE)

# Neural Multi-Task Recommendation from Multi-Behavior Data

Chen Gao*, Xiangnan He†, Dahua Gan*, Xiangning Chen*, Fuli Feng‡, Yong Li*, Tat-Seng Chua‡, Depeng Jin*
*Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
†School of Information Science and Technology, University of Science and Technology of China
‡School of Computing, National University of Singapore, Computing 1, Computing Drive, Singapore 117417
liyong07@tsinghua.edu.cn

***Abstract*—Most existing recommender systems leverage user behavior data of one type, such as the purchase behavior data in E-commerce. We argue that other types of user behavior data also provide valuable signal, such as views, clicks, and so on. In this work, we contribute a new solution named NMTR (short for Neural Multi-Task Recommendation) for learning recommender systems from user multi-behavior data. In particular, our model accounts for the cascading relationship among different types of behaviors (e.g., a user must click on a product before purchasing it). We perform a joint optimization based on the multi-task learning framework, where the optimization on a behavior is treated as a task. Extensive experiments on the real-world dataset demonstrate that NMTR significantly outperforms state-of-the-art recommender systems that are designed to learn from both single-behavior data and multi-behavior data.**

***Index Terms*—Multi-Behavior Recommendation, Collaborative Filtering, Deep Learning**

## I. Introduction

In online information systems, users interact with a system in a variety of forms. In traditional recommender systems, only user-item interaction data of one behavior type is considered for collaborative filtering [1]. Existing approaches for multi-behavior recommendation can be divided into two categories. The first category is based on collective matrix factorization (CMF) [2]–[4], which extends the matrix factorization (MF) method to jointly factorize multiple behavior matrices. The second category approaches the problem from the perspective of learning [5]–[7]. For example, [5], [7] extends the Bayesian Personalized Ranking (BPR) [8] framework to address multi-behavior recommendation by enriching the training data of relative preference from the multi-behavior data.

Despite effectiveness, we argue that existing models for multi-behavior recommendation suffer from three limitations.

- **Lack of behavior semantics**. Each behavior type has its own semantics and contexts, and more importantly, there exist strong ordinal relations among different behavior types. However, existing models have largely ignored the semantics of different behavior types.

忽略了不同行为的
不同语义

- **Unreasonable embedding learning**. The common setting of CMF is unreasonable. Specifically, a user's embedding vector represents his/her inherent interests, which should remain unchanged when the user performs different types of behaviors on items; and similarly for the item side.
- **Incapability in modeling complicated interactions**. Existing methods largely rely on MF to estimate a user's preference on an item. In MF, the interaction function is a fixed inner product, being insufficient to model the complicated and multi-type interactions between users and items.

To address the above mentioned limitations in multi-behavior recommendation, we propose a new solution named **N**eural **M**ulti-**T**ask **R**ecommendation (NMTR). Specifically, we separate the two components of embedding learning and interaction as advocated by the neural collbaborative filtering (NCF) framework [9]. We then design that 1) a user (and an item) has a shared embedding across multiple types of behaviors, and 2) a data-dependent interaction function is learned for each behavior type. Moreover, to incorporate the behavior semantics, especially the ordinal relation among behavior types, we relate the model prediction of each behavior type in a cascaded manner.

To summarize, the main contributions of this work are:

- We propose a novel neural network model tailored to learning user preference from multi-behavior data.
- To capture the ordinal relations among behavior types, we propose to correlate the model prediction of each behavior type in a cascaded way.
- Extensive experiments on the real-world dataset show that our method substantially outperforms existing methods.

## II. Problem Formulation

Let $\{\mathbf{Y}^1, \mathbf{Y}^2, ..., \mathbf{Y}^R\}$ denote the user-item interaction matrices for all the $R$ types of behaviors. Each interaction matrix is of size $M \times N$, where $M$ and $N$ denote the number of users and items, respectively. We assume that each entry of a interaction matrix has a value of 1 or 0:

$$y^r_{ui} = \begin{cases} 1, & \text{if } u \text{ has interacted with } i \text{ under behavior } r; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

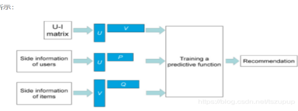
As we have discussed in the introduction, many user behavior types in real-world applications follow an ordinal (or sequential) relationship. Without loss of generality, we assume that the behavior types have a total order and sort them from the lowest level to the highest level: $\mathbf{Y}^1 \rightarrow \mathbf{Y}^2 ... \rightarrow \mathbf{Y}^R$, where $\mathbf{Y}^R$ denotes the target behavior to be optimized. An example of the target behavior is the purchase behavior in E-commerce, and other behaviors can include the click, adding to cart, etc.

The problem of multi-behavior recommendation is then formulated as follows.

**Input:** The user-item interaction data of the target behavior $\mathbf{Y}^R$, and the interaction data of other behavior types $\{\mathbf{Y}^1, \mathbf{Y}^2, ..., \mathbf{Y}^{R-1}\}$.

**Output:** A model that estimates the likelihood that a user $u$ will interact with an item $i$ under the target behavior.

CMF模型示意图如下所示:




CMF模型的公式化定义为:

The collective matrix factorization model extends the earlier model by making the user and item factor matrices also be able to make low-rank approximate factorizations of the user and item attributes:

$$\mathbf{X} \approx \mathbf{A}\mathbf{B}^T + \mu + \mathbf{b}_A + \mathbf{b}_B, \quad \mathbf{U} \approx \mathbf{A}\mathbf{C}^T + \mu_U, \quad \mathbf{I} \approx \mathbf{B}\mathbf{D}^T + \mu_I$$

Where

- $\mathbf{U}$ is the user attributes matrix, in which users are rows and attributes are columns.
- $\mathbf{I}$ is the item attributes matrix, in which items are rows and attributes are columns.
- $\mu_U$ are the column means for the user attributes (column vector).
- $\mu_I$ are the columns means for the item attributes (column vector).
- $\mathbf{C}$ and $\mathbf{D}$ are attribute-factor matrices (also model parameters).

2375-026X/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDE.2019.00140



IEEE computer society

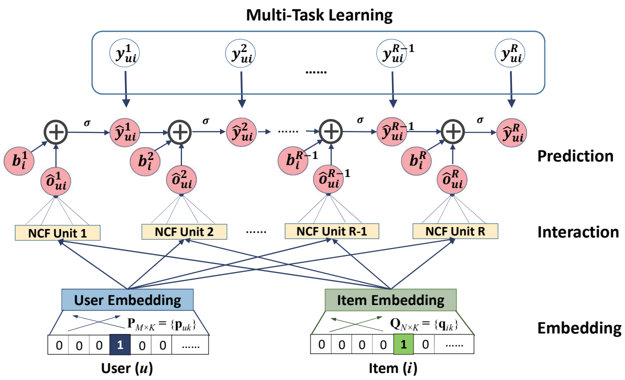



Fig. 1. Illustration of our proposed NMTR model.

## III. Method

Figure 1 illustrates our proposed NMTR model. Next, we elaborate the architecture component by component.

### A. Shared Embedding Layer

We apply one-hot encoding to encode the input of user ID and item ID. One advantage is that it can be easily extended to incorporate other features of a user and an item (e.g., user demographics and item attributes), if they are available in the application [10]. Let $\mathbf{v}_u^U$ and $\mathbf{v}_i^I$ denote the one-hot feature vector for user $u$ and item $i$. Then the embedding layer is defined as a linear fully connected layer without the bias terms:

$$\mathbf{p}_u = \mathbf{P}^T \mathbf{v}_u^U, \quad \mathbf{q}_i = \mathbf{Q}^T \mathbf{v}_i^I, \tag{2}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are the user embedding matrix and item embedding matrix, respectively. When only the ID feature is used to describe a user (or an item), $\mathbf{P}$ and $\mathbf{Q}$ are of the size $M \times E$ and $N \times E$, respectively, where $E$ denotes the embedding size; and $\mathbf{p}_u$ and $\mathbf{q}_i$ are essentially the $u-$th and $i-$th row vector of $\mathbf{P}$ and $\mathbf{Q}$, respectively.

### B. Separated Interaction Function

Above the embedding layer is the hidden layers that model the interaction between $\mathbf{p}_u$ and $\mathbf{q}_i$ to obtain the prediction score. Since we need to predict the likelihood of multiple behavior types with the same input, it is essential to learn a separated interaction function for each type. Let $f_\Theta^r$ denote the interaction function for the $r$-th type of behaviors with parameters $\Theta$, which outputs the likelihood that $u$ will perform a behavior of the $r$-th type:

$$\hat{y}_{ui}^r = \sigma(f_\Theta^r(\mathbf{p}_u, \mathbf{q}_i)), \tag{3}$$

where $\sigma$ denotes the sigmoid function converting the output to a probability. A good design of $f_\Theta^r$ is to have the ability and sufficient flexibility to learn the possible complicated patterns (e.g., collaborative filtering and others) in user behaviors. To achieve this, we consider the three neural network units proposed in the NCF paper [9], namely the generalized matrix factorization (GMF), multi-layer perceptron (MLP), and neural matrix factorization (NeuMF).

### C. Cascaded Predictions

Typically there are certain ordinal relations among behavior types in a real-world application, such as a user must view a product (i.e., click the product page) before she can purchase it. The existence of such relations implies that the predictive models for different behavior types should be related with each other, rather than being independent. To encode the sequential effect, we enforce that the prediction on a behavior type lies in the predictions of the precedent behavior types. Formally, we cascade the predictions of different behaviors as:

$$\begin{aligned}
\hat{y}_{ui}^R &= \sigma(\hat{y}_{ui}^{R-1} + f_\Theta^R(\mathbf{p}_u, \mathbf{q}_i) + b_i^R), \\
&\cdots\cdots \\
\hat{y}_{ui}^2 &= \sigma(\hat{y}_{ui}^1 + f_\Theta^2(\mathbf{p}_u, \mathbf{q}_i) + b_i^2), \\
\hat{y}_{ui}^1 &= \sigma(f_\Theta^1(\mathbf{p}_u, \mathbf{q}_i) + b_i^1),
\end{aligned} \tag{4}$$

where $b_i^r$ denotes the bias of item $i$ in the data of the $r$-th behavior type, and $f_\Theta^r$ denotes the interaction function for the $r$-th type of behaviors, which can be any of the three NCF units as introduced before. A graphical illustration of our cascading design can be found in the top part of Figure 1.

### D. Multi-Task Learning

As we have a dedicated model for each type of behaviors and the models follow a cascading prediction, it is intuitive to train models separately by following the order of $\hat{y}_{ui}^1, \hat{y}_{ui}^2, ..., \hat{y}_{ui}^R$. However, this way can not fully utilize the utility of learning from the data of multiple behaviors.

In contrast to training the models separately, multi-task learning (MTL) is a paradigm that performs joint training on the models of different but correlated tasks, so as to obtain a better model for each task [11]. The intuition for our design of cascaded predictions is that, if we can obtain improved models for other types of behavior, the model for the target behavior can also be improved. As such, we opt for MTL that trains all models simultaneously, where the model learning for each behavior type is treated as a task.

**Objective Function**. Following the probabilistic optimization framework [9], we obtain the loss function to be minimized as:

$$L = -\sum_{r=1}^{R} \lambda_r \Big( \sum_{(u,i) \in \mathcal{Y}_r^+} \log \hat{y}_{ui}^r + \sum_{(u,i) \in \mathcal{Y}_r^-} \log(1 - \hat{y}_{ui}^r) \Big), \tag{5}$$

where we additionally include the term $\lambda_r$ to control the influence of the $r$-th type of behaviors on the joint training. This is a hyper-parameter to be specified for different datasets, since the importance of a behavior type may vary for problems of different domains and scales. We additionally enforce that $\sum_{r=1}^{R} \lambda_r = 1$ to facilitate the tuning of these hyper-parameters.

**Training**. Since our model is composed of nonlinear neural networks, we optimize parameters with stochastic gradient descent (SGD), a generic solver for neural network models. Specifically, we utilize Adagrad as the optimizer and adopt L2 regularizer to solve over-fitting. As most machine learning toolkits (e.g., TensorFlow, Theano, PyTorch etc.) provide the function of automatic differentiation, we omit the derivation of the derivatives of our model.

## IV. Experiments

In this section, we conduct extensive experiments on the real-world dataset to answer the following research questions:

- **RQ1:** How does our proposed NMTR perform as compared with state-of-the-art recommender systems that are designed for learning from single-behavior and multi-behavior data?

TABLE I
STATISTICS OF OUR EVALUATION DATASET.

| Dataset | User# | Item# | Purchase# | Cart# | View# |
|---|---|---|---|---|---|
| Beibei | 21,716 | 7,977 | 295,622 | 642,622 | 2,412,586 |

- **RQ2:** How do the auxiliary behaviors affect NMTR's performance on the target behavior?
- **RQ3:** Can NMTR help to address the data sparsity problem, i.e., improving recommendations for sparse users with fewer interactions of the target behavior?

### A. Experimental Settings

*1) Dataset and Evaluation Protocol:* We experimented with a real-world E-commerce dataset that contains multiple types of user behaviors including purchases, views, adding to carts, etc. This dataset is collected from Beibei[1], the largest E-commerce platform for maternal and infant products in China. The statistics of the dataset are summarized in Table I. We applied the widely used leave-one-out technique and then adopted two popular metrics, *HR* (Hit Ratio) and *NDCG* (Normalized Discounted Cumulative Gain), to judge the performance of the ranking list.

*2) Baselines:* We compared the performance of our proposed NMTR with 9 baselines. The compared single-behavior methods are introduced as follows.

**BPR [8]** *Bayesian Personalized Ranking* (BPR) is a widely used pairwise learning framework for item recommendation with implicit feedback.

**NCF [9]** *Neural Collaborative Filtering* (NCF) is a neural framework to learn interactions between the latent features of users and items. There are some variants: **GMF**, **MLP** and **NeuMF**.

The second group of five compared methods that can leverage multiple types of behavior data are as follows.

**CMF [4]** CMF decomposes the data matrices of multiple behavior types simultaneously.

**MC-BPR [5]** Multi-Channel BPR [5] is the state-of-the-art solution for multi-behavior recommendation. It adapts the negative sampling rule in BPR for multi-behavior data.

**MC-NCF** We replaced the basic MF model in MC-BPR with the NCF unit, and named three variants as **MC-GMF**, **MC-MLP** and **MC-NeuMF**.

*3) Parameter Settings:* We implemented our NMTR and baseline methods in TensorFlow[2]. Since we have three choices of NCF units as the interaction function, we name the respective methods as **NMTR-GMF**, **NMTR-MLP** and **NMTR-NeuMF**. We randomly selected a training instance for each user as the validation set to tune hyper-parameters.

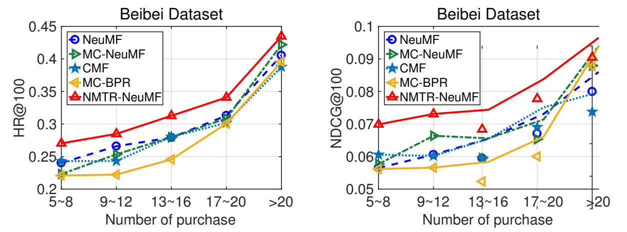
### B. Performance Comparison (RQ1)

We first compare the top-K recommendation performance with state-of-the-art methods. We investigate the top-$K$ performance with $K$ setting to [50, 100]. Note that for a user, our evaluation protocol ranks all unobserved items in the training set [12].

Table II shows the performance of HR@K and NDCG@K for our three NMTR methods, five multi-behavior recommendation methods, and four single-behavior methods. From the

[1]https://www.beibei.com

[2]https://www.tensorflow.org




(a) HR@100 (b) NDCG@100

Fig. 2. Performance of NeuMF, MC-NeuMF, CMF, MC-BPR and NMTR-NeuMF on users with different number of purchase records

results, we observe that our proposed NMTR methods obtain the best performance in terms of HR@K and NDCG@K as compared to all baselines. The one-sample paired t-tests indicate that all improvements are statistically significant for $p < 0.05$. Among the three NMTR methods, NMTR-GMF and NMTR-NeuMF are better than NMTR-MLP, which verifies the effectiveness of the element-wise operator in learning the user-item interaction function. Compared with the best single-behavior baseline NeuMF, NMTR outperforms it by 9.01% in HR and 6.72% in NDCG on the Beibei dataset. Compared with MC-NeuMF, which extends NeuMF on multi-behavior data with the Multi-Channel BPR [5], NMTR obtains an improvement in HR of 6.08%.

### C. Impact of Auxiliary Behaviors(RQ2)

We investigate how the data quality of auxiliary behaviors affects our NMTR model's performance. A intuitive experimental setting is that to random sample auxiliary behaviors for our utilized dataset while keeping target behavior (i.e. purchase) intact. Table III shows the performance of different combinations of behavioral data. From the results, we have the following two observations. First, adding views data leads to better performance than adding carts data. The main reason is probably that the cart data contains too similar signal with the purchase data and provides fewer new signal on user preference. Specifically, a purchase record is often accompanied by a carting record. Second, by using only 50% of the cart and view interactions, we find that the performance is worse than the previous two experiments. Specifically, the performance of (Purchase, 50% Carting) is worse than only using purchase, while (Purchase, 50% Viewing) is better than only using purchase. There are two major reasons. On one hand, view is the weakest signal to reflect user preference and the total number of views is very large, making the missing of part of view data is acceptable. Therefore, missing of some view records shall not affect the result too much. On the other hand, random missing of carts records can bring some noises.

### D. Impact of Data Sparsity (RQ3)

We further study how our proposed NMTR model improves the recommendation for those users having few records of target behavior. eliminates the randomness of experimental results. The results are shown in Figure 2. From the results, we can observe that when the user purchase data becomes sparser, the recommendation performance of NMTR-NeuMF decreases slower than other methods. Especially for NDCG, from fifth to first user group, NMTR-NeuMF is decreased by 27.56% while MC-BPR and MC-NeuMF is decreased by 40.09% and 38.62%. Furthermore, even in the first user group with only

TABLE II
TOP-K RECOMMENDATION PERFORMANCE COMPARISON (K IS SET TO 50, 80, 100, 200)

| | | Beibei Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Group** | **Method** | **HR@50** | **NDCG@50** | **HR@80** | **NDCG@80** | **HR@100** | **NDCG@100** | **HR@200** | **NDCG@200** |
| Our NMTR Model | NMTR-GMF | 0.2050 | 0.0590 | **0.2721** | **0.0688** | 0.3119 | 0.0741 | 0.4543 | 0.0961 |
| | NMTR-MLP | 0.1928 | 0.0560 | 0.2690 | 0.0676 | 0.3188 | **0.0762** | 0.4732 | 0.0967 |
| | NMTR-NeuMF | **0.2079** | **0.0609** | 0.2689 | 0.0683 | **0.3193** | 0.0760 | **0.4766** | **0.0971** |
| Multi-behavior | CMF | 0.1596 | 0.0481 | 0.2377 | 0.0606 | 0.2829 | 0.0663 | 0.4191 | 0.0850 |
| | MC-BPR | 0.1743 | 0.0503 | 0.2299 | 0.0604 | 0.2659 | 0.0647 | 0.3852 | 0.0786 |
| | MC-GMF | 0.1822 | 0.0508 | 0.2425 | 0.0611 | 0.2975 | 0.0690 | 0.4262 | 0.0891 |
| | MC-MLP | 0.1810 | 0.0534 | 0.2342 | 0.0598 | 0.2810 | 0.0684 | 0.4116 | 0.0834 |
| | MC-NeuMF | 0.2014 | 0.0577 | 0.2522 | 0.0669 | 0.3010 | 0.0719 | 0.4300 | 0.0897 |
| Single-behavior | BPR | 0.1199 | 0.0348 | 0.1686 | 0.0419 | 0.2002 | 0.0463 | 0.3039 | 0.0624 |
| | GMF | 0.1792 | 0.0475 | 0.2555 | 0.0608 | 0.2920 | 0.0665 | 0.4090 | 0.0828 |
| | MLP | 0.1711 | 0.0483 | 0.2383 | 0.0459 | 0.2679 | 0.0617 | 0.3947 | 0.0792 |
| | NeuMF | 0.1828 | 0.0573 | 0.2559 | 0.0668 | 0.2929 | 0.0714 | 0.4078 | 0.0852 |

TABLE III
PERFORMANCE OF NMTR MODEL WITH DIFFERENT COMBINATION OF INTERACTION DATA

| | Beibei Dataset | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Interaction Subset** | **(Purchase, Carting)** | | **(Purchase, View)** | | **(Purchase, 50% Carting)** | | **(Purchase, 50% View)** | |
| **Performance** | **HR@100** | **NDCG@100** | **HR@100** | **NDCG@100** | **HR@100** | **NDCG@100** | **HR@100** | **NDCG@100** |
| **NMTR-GMF** | 0.2979 | 0.0705 | 0.3029 | 0.0726 | 0.2947 | 0.0701 | 0.2953 | 0.0698 |
| **NMTR-MLP** | 0.2770 | 0.0670 | 0.3140 | 0.0741 | 0.2726 | 0.0654 | 0.3058 | 0.0725 |
| **NMTR-NeuMF** | 0.2882 | 0.0691 | 0.3147 | 0.0743 | 0.2778 | 0.0676 | 0.3107 | 0.0737 |

5-8 purchase records, our NMTR still keeps a good recommendation performance of 0.027 for HR@100 and 0.07 for NDCG@100, which outperforms the best baseline by 11.23% and 15.35%, respectively. As a result, the performance gap between NMTR and other methods becomes larger when data become sparser. As a summary, we conclude that our proposed NMTR model alleviate data sparsity problem efficiently to some extent.

## V. RELATED WORK

Multi-behavior based recommendation aims to leverage the behavior data of other types to improve the recommendation performance on the target behavior. Matrix factorization, a prevalent method for single-behavior based recommendation [8], [12], [13], has been adapted to the multi-behavior scenario. Ajit *et al.* [2] first proposed a collective matrix factorization model (CMF) to simultaneously factorize multiple user-item interaction matrices with sharing item-side embeddings across matrices. Some other works extended the CMF to handle datasets of multiple user behaviors [3], [4]. Zhe *et al.* [4] considered different behaviors in online social network (comment, re-share, and create-post), while Artus *et al.* [3] extended CMF with sharing user-side embeddings in recommendation based social network data.

On the other hand, some works approach multi-behavior recommendation from the perspective of learning [5]–[7]. Babak *et al.* [5] proposed an extension of Bayesian Personalized Ranking (BPR) [8], as Multi-channel BPR, to adapt the sampling rule from different types of behavior in training of standard BPR. Recently, Ding *et al.* [7] assign different weights to multiple types of behaviors in the training of matrix factorization with considering some specific behaviors, such as view but not purchase. Later on, the same authors [6] developed a margin-based pairwise learning framework to jointly capture observed multiple types of behaviors and unobserved interactions.

As discussed in the introduction, these existing models suffer from several limitations, which are addressed by our neural network-based solution NMTR.

## VI. CONCLUSION

In this work, we designed a recommendation system to exploit multiple types of user behaviors. We proposed a neural network method named NMTR, which combines the recent advances of NCF modeling and the efficacy of multi-task learning. We conducted extensive experiments on the real-world dataset and demonstrated the effectiveness of our NMTR method on multiple recommender models. This work makes the first step towards understanding how to integrate the rich semantics of users' multiple behaviors into recommender systems. With increasing kinds of user behaviors on the Web, we believe multi-behavior recommendation is an important topic and will attract more attention in the future.

## REFERENCES


[1] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *SIGIR*, 2016, pp. 549–558.
[2] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *SIGKDD*. ACM, 2008, pp. 650–658.
[3] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *WSDM*, 2012, pp. 173–182.
[4] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," in *WWW*, 2015, pp. 1406–1416.
[5] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Bayesian personalized ranking with multi-channel user feedback," in *RecSys*, 2016, pp. 361–364.
[6] J. Ding, G. Yu, X. He, Y. Quan, Y. Li, T.-S. Chua, D. Jin, and J. Yu, "Improving implicit recommender systems with view data." in *IJCAI*, 2018, pp. 3343–3349.
[7] J. Ding, F. Feng, X. He, G. Yu, Y. Li, and D. Jin, "An improved sampler for bayesian personalized ranking by leveraging view data," in *WWW*, 2018, pp. 13–14.
[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.
[9] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.
[10] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua, "Tem: Tree-enhanced embedding model for explainable recommendation," in *WWW*, 2018, pp. 1543–1552.
[11] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *SIGKDD*, 2004, pp. 109–117.
[12] S. Kabbur, X. Ning, and G. Karypis, "Fism: factored item similarity models for top-n recommender systems," in *SIGKDD*, 2013, pp. 659–667.
[13] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *SIGIR*, 2018, pp. 355–364.