Hindawi
Complexity
Volume 2020, Article ID 8828087, 13 pages
https://doi.org/10.1155/2020/8828087

Research Article

Deep Interest-Shifting Network with Meta-Embeddings for Fresh Item Recommendation

Zhao Li,[1] Haobo Wang,[2] Donghui Ding,[1] Shichang Hu,[1] Zhen Zhang,[2] Weiwei Liu,[3] Jianliang Gao,[4] Zhiqiang Zhang,[5] and Ji Zhang[6]

[1]*Alibaba Group, Hangzhou, China*
[2]*Zhejiang University, Hangzhou, China*
[3]*Wuhan University, Wuhan, China*
[4]*Central South University, Changsha, China*
[5]*Zhejiang University of Finance and Economics, Hangzhou, China*
[6]*University of Southern Queensland, Toowoomba, Queensland, Australia*

Correspondence should be addressed to Jianliang Gao; gaojianliang@csu.edu.cn

Received 18 August 2020; Revised 22 September 2020; Accepted 8 October 2020; Published 28 October 2020

Academic Editor: Jia Wu

Nowadays, people have an increasing interest in fresh products such as new shoes and cosmetics. To this end, an E-commerce platform Taobao launched a fresh-item hub page on the recommender system, with which customers can freely and exclusively explore and purchase fresh items, namely, the *New Tendency* page. In this work, we make a first attempt to tackle the fresh-item recommendation task with two major challenges. First, a fresh-item recommendation scenario usually faces the challenge that the training data are highly deficient due to low page views. In this paper, we propose a deep interest-shifting network (DisNet), which transfers knowledge from a huge number of auxiliary data and then shifts user interests with contextual information. Furthermore, three interpretable interest-shifting operators are introduced. Second, since the items are fresh, many of them have never been exposed to users, leading to a severe cold-start problem. Though this problem can be alleviated by knowledge transfer, we further babysit these fully cold-start items by a relational meta-Id-embedding generator (RM-IdEG). Specifically, it trains the item id embeddings in a learning-to-learn manner and integrates relational information for better embedding performance. We conducted comprehensive experiments on both synthetic datasets as well as a real-world dataset. Both DisNet and RM-IdEG significantly outperform state-of-the-art approaches, respectively. Empirical results clearly verify the effectiveness of the proposed techniques, which are arguably promising and scalable in real-world applications.

1. Introduction

E-commerce has been prevalent in our daily life. In traditional online shopping scenarios, all items are mixed up, and a recommender system predicts users' preferences on items based on their past interactions, e.g., click, purchase, and rating [1–3]. However, this strategy overlooks the influence of the items' life periods and causes two problems. First, as many people have a growing interest in novel, newly released commodities, their requirements will not be fully satisfied. Second, popular items have more opportunities to be exposed, whereas those new products are overwhelmed, even though with high quality [4–6].

To tackle these problems, one E-commerce platform Taobao launched a new application, namely, *New Tendency* page, aiming to recommend fresh items for users who prefer new products. As illustrated in Figure 1, a card which contains a fresh item with its textual descriptions is pushed to the users. Once a user clicks this card, the *New Tendency* page appears, where more items from a predefined fresh item pool are recommended to this user. As a result, users who prefer newly released products can freely explore this

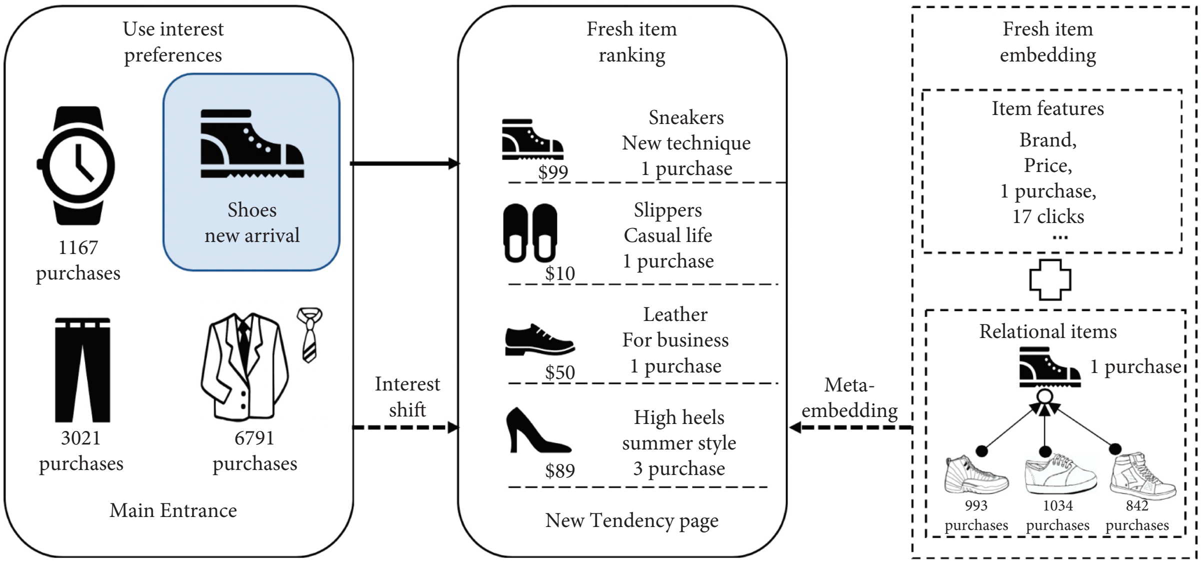


Figure 1: An illustration of fresh item recommended. Once a user clicks the card on the left side, a fresh-item recommendation page (middle side) appears to achieve interest shifting from users' history interests to the fresh items, where the recommended fresh items are ranked using the embeddings of fresh items and relational items (right side). All the fresh items on the recommended page are chosen from a specific fresh item pool, which are usually less exposed than items on the main entrance page.

page. However, to achieve high-quality ranking on this page, two key problems have to be addressed.

1.1. Q1: How to Address the Data-Deficiency Problem? Recommending fresh items directly on the main entrance page of the app may cause unpredictable influences. Thus, this page has to be designed as a fresh-item recommendation scenario. Compared to the main entrance page of the app, the *New Tendency* page is reported to contain less than 5% of page views. Most of the fresh items only have a few interactions, which make the scenario-specific training data highly deficient. As a result, we have to collect additional information to improve the performance.

1.2. Potential Solutions to Q1. We firstly notice that the clicked card contains rich contextual information, such as the showpiece and its textual description, which clearly reflects the user interests. Therefore, we can utilize off-the-shelf context-aware recommender systems (CARSs) [7], such as factorization-based approaches [8, 9] and deep learning-based models [10–15]. However, the model complexity increases owing to the involvement of the context features, which prevent the model being trained sufficiently. To deal with this problem, cross-domain recommender systems (CDRSs) [16–18] seem appealing due to their superiority in handling data deficiency. In particular, an asymmetric CDRS [19–21], which collects a large amount of context-free data (e.g., data from the main entrance of the app, namely, auxiliary data), can be designed to improve the prediction performance. However, existing asymmetric CDRS models seldom consider the scenario-specific contextual information of the target domain.

1.3. Q2: How to Deal with Totally Cold-Start Items? As reported by Taobao, more than 60% of fresh items are newborn and never interacted by users, which causes a severe cold-start problem. Note that these newborn items are not the cause of data deficiency because they are not a part of training data.

1.4. Potential Solutions to Q2. The cold-start problem is usually solved by integrating external information, e.g., item attributes [22, 23], user attributes [24, 25], relational data [26], and knowledge from other domains [16]. We note that this problem can be alleviated by applying the cross-domain technique because the embeddings of item attributes can be reused. Nevertheless, since the id of a cold-start item never appears, its embedding cannot obtain a good initialization. Pan et al. [27] proposed the meta-Id-embedding generator (Meta-IdEG), which considers the id embedding initialization problem and solves it through a learning-to-learn training manner. However, meta-IdEG only utilizes item features to generate the id embedding. As a result, it is unable to explore the community structural information when initializing id embeddings, which leads to a suboptimal solution.

1.5. Our Solutions. In this study, we propose two novel techniques to construct a deep learning-based recommender system, which simultaneously tackles these above issues. The proposed model fully exploits various types of external information to improve the prediction performance. To answer Q1, we present a deep interest-shifting network (DisNet). Specifically, it firstly learns the users' general interest vectors using a huge number of auxiliary data and then shifts them to a scenario-specific representation using contexts. Next, the size of trainable parameters is reduced to a few neural network layers, which significantly alleviates the data-deficiency problem. To answer Q2, the transferred embedding layer of item attributes can be reused, and the only thing that matters is the item id embedding initialization problem. Hence, this paper proposes a relational meta-Id-embedding generator (RM-IdEG), which is trained in a learning-to-learn manner, aiming to make the model achieve great generalization ability after few-shot training. Furthermore, RM-IdEG absorbs the information of relevant items. Therefore, the community structural information can be inherently embedded and exploited, which has been proved beneficial for addressing the cold-start problem [26].

The main contributions of this work are summarized as follows:

> A novel application, fresh item recommendation, is studied, which gives new items more opportunities to be exposed and fully personalizes the recommendations of those who prefer the novel, innovative products. We also make a first attempt to address the fresh-item recommendation task by two novel techniques.
>
> We present a deep interest-shifting network (DisNet) to deal with the severe data-deficiency problem in a fresh-item recommendation scenario.
>
> To address the cold-start problem, we propose a relational meta-Id-embedding generator (RM-IdEG) that involves the relational data into meta-id embedding initialization, which enables community structural information to be inherently contained.
>
> Extensive experimental results demonstrate that our model can effectively handle fresh-item recommendation tasks in both cold-start and warm-start stages.

The rest of this work is organized as follows. In the next section, notations and preliminary knowledge are introduced. In Section 3, we provide a detailed description of our network architecture. After that, the results of empirical studies are reported. Then, we give the related works of our method. Further discussion and concluding remarks are provided in the last section.

2. Notations and Preliminaries

In this section, we firstly discuss a popular architecture of context-aware recommender systems. Then, we introduce the training procedure of meta-IdEG and summarize the notations in Table 1.

2.1. Context-Aware Recommendation. A popular strategy in existing context-aware recommendation systems is to learn latent representations for users and items and then make decisions using these latent vectors.

Formally, given an example, which contains an item t, a user u, and potentially some contexts, we first feed them into an embedding layer. Then, their features are transformed into vector representations by one-hot encoding or multihot encoding. The transformed item features consist of an item id embedding $\mathbf{e}_t$ and other content features $\mathbf{v}_t$. For the user, we combine its id embedding and other features as one vectorized representation $\mathbf{v}_u$. Finally, we denote the transformed context features by $\mathbf{c}$. The final prediction is made by

$$\begin{aligned} \hat{y} &= g(\mathbf{q}_\mathbf{u}, \mathbf{p}_t, c), \\ \mathbf{q}_\mathbf{u} &= f_\mathbf{u}(\mathbf{v}_\mathbf{u}), \\ \mathbf{p}_t &= f_t(\mathbf{e}_t, \mathbf{v}_t). \end{aligned} \tag{1}$$

For example, in matrix factorization-based models [28], $\mathbf{q}_u$ and $\mathbf{p}_t$ are exactly their id embeddings, and g is the context-biased prediction function. State-of-the-art models [29, 30] also use neural networks to learn user/item representation as well as make decisions. This paper also adopts neural networks for f_t, f_u, and g, which lead to a double-tower model architecture.

It is noteworthy that such a learning paradigm deeply couples the contextual information in the model architecture. In our cross-domain setting, there are heterogeneous contexts, i.e., scenario-specific contexts. Therefore, the trainable parameters of deep neural network models cannot be reused, which makes them hard to share knowledge across different domains [31–33].

2.2. Meta-Id Embedding Generator. To babysit newborn items, the only thing that matters is how to learn the embeddings for new items' ids. A common learning paradigm first uses an Id embedding generator (IdEG) to initialize a vector for new ids in the embedding table and then update them using incoming user interactions. The most intuitive way is to output a random embedding initialization. However, its generalization ability may be restricted due to the cold-start problem. To this end, Pan et al. [27] proposed to initialize id embeddings using meta-learning technique, a.k.a. meta-Id embedding generator (meta-IdEG). By regarding the recommendation for each item as a task, meta-IdEG ensures good embedding initialization such that the model achieves better generalization ability after few-shot training.

Next, we illustrate the workflow of meta-IdEG. For each task that relates to a specific item, we divide its data examples (interactions) into two sets: a support set $\mathcal{D}^s$ and a query set $\mathcal{D}^q$. We firstly feed the item features into a neural network to generate an id embedding, $\mathbf{e}_t^* = \mathrm{IdEG}_{\mathrm{meta}}(\mathbf{v}_t)$. Then, we optimize $\mathrm{IdEG}_{\mathrm{meta}}$ in a learning-to-learn manner. We denote the predicted label on the support set as $\hat{y}^*$ using $\mathbf{e}_t^*$. First, we can obtain the cold-start loss by

$$l_c(y, \hat{y}^*; \mathbf{e}_t^*, \mathcal{D}^s) = -y\log(\hat{y}^*) - (1-y)\log(1-\hat{y}^*). \tag{2}$$

Then, we update the embedding by one step of gradient descent:

$$\mathbf{e}_t' = \mathbf{e}_t^* - \alpha\frac{\partial l_c}{\partial \mathbf{e}_t^*}, \tag{3}$$

where α is the learning rate. Since a new embedding is obtained, we can predict label $\hat{y}'$ on the query set using $\mathbf{e}_t'$. Next, we define a warmed loss by

$$l_w(y, \hat{y}'; \mathbf{e}_t', \mathcal{D}^q) = -y\log(\hat{y}') - (1-y)\log(1-\hat{y}'). \tag{4}$$

Table 1: Notations.

Notation	Definition or descriptions
(u, t)	A pair of user u and item t
$\mathbf{e}_t$	The id embedding of item t
$\mathbf{v}_t$	The embedded features (except id) of item t
$\mathbf{v}_u$	The embedded features (include id) of user u
$\mathbf{c}$, $\mathbf{c}_s$	Common contexts and scenario-specific contexts
f_t, f_u	Item/user representation networks
g	Decision-making network
h	Context network
$\mathbf{q}_u$, $\mathbf{p}_t$	Latent user/item vectors in the interest space
$\mathbf{z}_s$, $\mathbf{Z}_s$	Output vector/matrix of context network h
$\mathbf{e}_\mathbf{t}^*$, $\mathbf{e}_\mathbf{t}'$	Cold-start and warmed item id embeddings in the meta-training procedure
$\mathcal{D}^s$, $\mathcal{D}^q$	Support and query sets of a cold-start item in $\mathcal{D}^c$
$\hat{y}^*$, $\hat{y}'$	Predicted labels on $\mathcal{D}^s$ and $\mathcal{D}^q$
l_c, l_w, l_{meta}	Cold-start, warmed, and meta-loss in the meta-training procedure
$\mathcal{D}^a$, $\mathcal{D}^w$, $\mathcal{D}^c$	Auxiliary, warm-start, and cold-start datasets
y, $\hat{y}$, $\hat{y}^{\text{DN}}$	Ground-truth label and predicted labels of context-aware models and DisNet
$\text{ISO}(\cdot)$	Interest-shifting operator
$\mathbf{e}_\mathbf{t}^\mathbf{i}$ $(i = 1, \ldots, k)$	The id embeddings of top k relevant items
$\mathbf{a}_\mathbf{t}^\mathbf{i}$ $(i = 1, \ldots, k)$	Attention scores
$\mathbf{W}_i$, $\mathbf{b}_i$ $(i = 1, 2)$	Weight matrices and bias vectors of the NN operator
$\widetilde{\mathbf{W}}_i$, $\widetilde{\mathbf{b}}_1$ $(i = 1, 2)$	Weight matrices and bias vector of RM-IdEG
$\mathbf{W}$, $\mathbf{h}$, $\mathbf{b}$	Parameters of attentional embedding aggregator
$\mathcal{T}$, $\mathcal{T}^c$	Item sets of $\mathcal{D}^w \cup \mathcal{D}^c$ and $\mathcal{D}^c$

Note that $\mathbf{e}_t^*$ and $\mathbf{e}_t'$ do not have to be explicitly computed, and we are only interested in their gradients on $\text{IdEG}_{\text{meta}}$. Finally, we sum the two losses to get our meta-loss function:

$$l_{\text{meta}} = l_c + \eta l_w. \tag{5}$$

Here, η is the tradeoff parameter. In other words, minimizing l_{meta} simultaneously achieves two goals: (1) the error in predictions for the new items should be small; (2) after a small amount of labeled data is collected, a few gradient descent updates should lead to good generalization ability.

3. Proposed Model

3.1. Deep Interest-Shifting Network. In this section, we present DisNet, a learning framework for recommending items in a fresh-item recommendation page, which usually contains rich scenario-specific contexts. The overall network architecture is shown in Figure 2.

We note that the latent vector of a user actually reflects his or her interest in a latent space, while the scenario-specific contexts reflect the interest shifting in the user's general interests [34, 35]. For example, there is a boy who is interested in sports, games, and electronic products. Once he clicks a fresh item iPhone-11, he may pay more attention to electronic products with advanced technology, and we can recommend him newly released smartphones, laptops, and so on. We assume that such interest shifting will not change its latent semantics. In other words, the shifted representations can directly be fed into the decision-making network g. By this assumption, we can decouple the general interest of users from the scenario-specific contexts. Denoting the scenario-specific context by $\mathbf{c}_s$, we propose an interest-shifting operator (ISO) to obtain a shifted user representation:

$$\begin{aligned} \mathbf{q}_u^s &= \text{ISO}\left(\mathbf{q}_u, \mathbf{z}_s\right), \\ \mathbf{z}_s &= h\left(\mathbf{c}_s\right), \end{aligned} \tag{6}$$

where $\mathbf{q}_u^s$ and $\mathbf{q}_u$ have the same dimension m. h maps the contexts to a latent space to extract their critical information.

It is noteworthy that there is a huge amount of auxiliary data, from which we can model the general interest of the users. Thus, we can pretrain the item/user representation networks as well as the decision-making network using these data. We denote the pretrained networks by $\overline{f}_t$, $\overline{f}_u$, and $\overline{g}$. Then, the context information can be incorporated to shift the latent user vector to a scenario-specific one but in the same interest space. Formally, DisNet makes the decision by

$$\hat{y}^{\text{DN}} = \overline{g}\left(\mathbf{q}_u^s, \mathbf{p}_t, \mathbf{c}\right) = \overline{g}\left(\text{ISO}\left(\overline{f}_u\left(\mathbf{v}_u\right), \mathbf{z}_s\right), \overline{f}_t\left(\mathbf{e}_t, \mathbf{v}_t\right), \mathbf{c}\right). \tag{7}$$

Such a model not only transfers knowledge from a general interest domain that has rich data samples but also reduces the size of the trainable parameters to the $\text{ISO}(\cdot)$ and h functions only. Obviously, the context-aware and data-deficiency problems can be addressed simultaneously.

Note that c is some contexts shared by the two domains. However, it is possible that auxiliary data have their own context as well. We ignore such contextual information and preserve the common parts only because we are modeling the general interest of the users. In practice, we also enable the decision-making network $\overline{g}$ and the embedding layer to be fine-tuned.

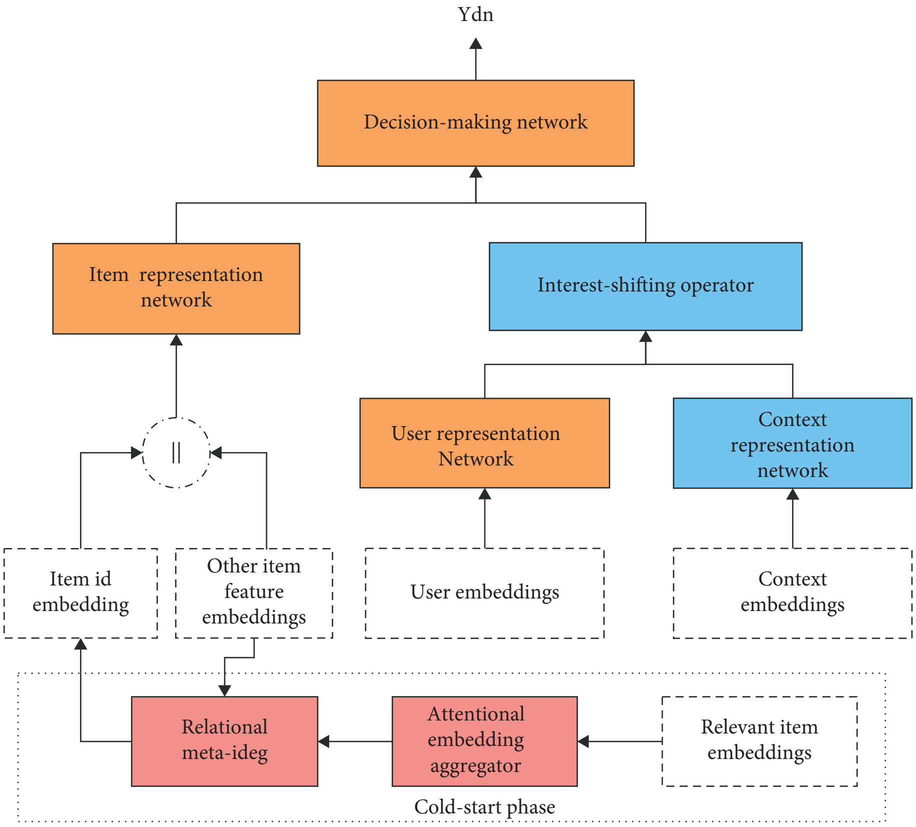

Ydn
Decision-making network
Item representation network
Interest-shifting operator
User representation Network
Context representation network
||
Item id embedding
Other item feature embeddings
User embeddings
Context embeddings
Relational meta-ideg
Attentional embedding aggregator
Relevant item embeddings
Cold-start phase

Figure 2: The model architecture of DisNet and RM-IdEG. The orange parts are pretrained using the auxiliary dataset. The blue ones are trained using the warm-start dataset. The pink parts are trained using the cold-start dataset.

3.1.1. *Interest-Shifting Operators*.

The above discussion provides the overall network architecture. Now, we can perform any reasonable shifting operation to learn the context-specific representation of the user. In this work, we introduce three interest-shifting operators, all of which relate to very interesting interpretations.

Add Operator. Motivated by the huge success of the representation learning and knowledge graph, we adopt a similar strategy as TransR [36]. Specifically, it embeds each entity and relation by optimizing the translation principle $\mathbf{e}_a + \mathbf{e}_r \approx \mathbf{e}_b$ if a triplet (a, r, b) exists in the graph. Recall the example of interest shifting, i.e., when a boy clicks an item iPhone-11, the interest representation of this boy goes to the interest of a boy who has a preference for electronic products with advanced technology. If we regard the contextual information as a relation, we obtain our first operator, which adds up the latent user vector and contextual vector:

$$\mathrm{ISO}(\mathbf{q}_u, \mathbf{z}_s) = \mathbf{q}_u + \mathbf{z}_s. \tag{8}$$

This implies that $\mathbf{q}_u$ and $\mathbf{z}_s$ have the same dimension. That is, the projection function h directly learns the discrepancy between the original interest and the shifted interest, which is similar to the relation embedding in the knowledge graph.

COT Operator. Before introducing the second operator, we review a popular technique in the context-aware recommendation, namely, the contextual operation tensor (COT) [37]. By estimating a contextual operation matrix, COT maps the original user/item latent vectors to their context-specific ones. We notice that COT has three main limitations: (1) it assumes the context space is fixed and the contextual operation matrix relates to different context values; (2) it jointly learns the original latent vectors as well as the contextual operation matrix; and (3) it uses linear mapping, i.e., a 3D tensor, to obtain the contextual operation matrix, which leads to degenerated performance. Obviously, COT cannot be applied to our problem directly because the data-deficiency problem prevents the joint learning procedure, and cross-domain data have different contexts.

Fortunately, in DisNet, we have decoupled the user's general interest from the scenario-specific interest. Therefore, we can estimate the scenario-specific context operation matrix using the h function:

$$\begin{aligned} \mathrm{ISO}(\mathbf{q}_u, \mathbf{Z}_s) &= \mathbf{Z}_s^\top \mathbf{q}_u, \\ \mathbf{Z}_s &= h(\mathbf{c}_s). \end{aligned} \tag{9}$$

Here, h outputs a $d \times d$ matrix instead of a single vector. In other words, while COT focuses on different context values, our model considers how external contexts affect the user's interest.

Neural Network-Based Operator. Yet, we have only considered linear shifting, while in reality, the transformation may be nonlinear. To bridge this gap, we propose a neural network-based operator:

$$\mathrm{ISO}\left(\mathbf{q}_u, \mathbf{z}_s\right) = \mathbf{W}_2^{\top} \sigma\left(\mathbf{W}_1^{\top}\left[\mathbf{q}_u \| \mathbf{z}_s\right] + \mathbf{b}_1\right) + \mathbf{b}_2, \tag{10}$$

where $\mathbf{W}_i, \mathbf{b}_i$ $(i \in \{1, 2\})$ refer to the weight matrices and bias vectors. σ is the activation function. $[\cdot\|\cdot]$ denotes the concatenation of two vectors. It is worth pointing out that any network architectures can be used, and this paper considers a simple multilayer perceptron.

While the add operator regards the contexts as bias and the COT operator considers the cross-influences between the user interest and contexts, the NN-based operator achieves these two goals simultaneously.

3.2. Relational Meta-Id-Embedding Generator. This section concentrates on babysitting fresh items in the cold-start phase, where they suffer from a severe cold-start problem. It is worth noting that DisNet can reuse the embedding layer after pretraining. Then, all the attributes except item id obtain great embeddings. Hence, the only thing that matters is the item id embedding initialization. Following [27], this work learns an IdEG in a learning-to-learn manner. Nevertheless, we notice that the vanilla meta-IdEG feeds item features into a simple neural network to generate embeddings. Obviously, meta-IdEG neglects the fact that id embedding reflects the community structural information between items, exploiting which has been proved beneficial for alleviating the cold-start problem [26].

To remedy this problem, a novel relational meta-Id embedding generator (RM-IdEG) is proposed, whereas it trains the item id embedding in a learning-to-learn manner and integrates relational information for better embedding initialization, which further improves the performance of DisNet on new items. Specifically, we collect a set of warm-start items that are significantly relevant to the cold-start item t. Many influential relations can be considered, such as items from the same seller and the same brand. For instance, a newly released Nike T-shirt may have similar selling behaviors as other items in Nike shops. Then, we construct an id embedding set $\mathcal{I}_t = \{\mathbf{e}_t^1, \ldots, \mathbf{e}_t^k\}$. Here, $\mathbf{e}_t^i$ $(i = 1, 2, \ldots, k)$ denote the id embeddings of top k relevant items. Then, we output the new embedding via an attentional embedding aggregator:

$$\begin{aligned} \mathbf{e}_t^* &= \frac{1}{C} \sum_{i=1}^{k} a_t^i \mathbf{e}_t^i, \\ C &= \sum_{i=1}^{k} a_t^i. \end{aligned} \tag{11}$$

Here, C is used for normalization. The attention score a_t^i is given by a global attention network:

$$a_t^i = \exp\left(\mathbf{h}^T \sigma\left(\mathbf{W}^{\top} \mathbf{e}_t^i + \mathbf{b}\right)\right), \tag{12}$$

where $\mathbf{h}$, $\mathbf{W}$, and $\mathbf{b}$ are shared attention parameters. Then, we feed the learned attentional id embedding and item features into a neural network to obtain the final embedding:

$$\mathbf{e}_t = \underset{\mathrm{rm}}{\mathrm{IdEG}}\left(\mathbf{v}_t, \mathcal{I}_t\right) = \tanh\left(\widetilde{\mathbf{W}}_2^{\top} \sigma\left(\widetilde{\mathbf{W}}_1^{\top}\left[\mathbf{e}_t^* \| \mathbf{v}_t\right] + \widetilde{\mathbf{b}}_1\right)\right), \tag{13}$$

where $\widetilde{\mathbf{W}}_i$ $(i \in \{1, 2\})$ are weight matrices and $\widetilde{\mathbf{b}}_1$ is the bias vector. To obtain numerically stable outputs, we follow some tricks in [27]: (1) the bias of the last layer is removed; (2) tanh activation is applied in the final layer.

Remark 1. The proposed model fully addresses the cold-start problem from two aspects: (1) through a learning-to-learn training procedure, our model achieves better generalization ability with few training data; (2) by considering influentially relevant items, RM-IdEG automatically encodes community structural information into the embedding initialization, and the predictive accuracy is further improved.

3.3. Training. Now, we describe the training procedure of our model. Note that the training fresh item set $\mathcal{T}$ does not contain those newborn items. Consequently, we choose an item subset $\mathcal{T}^c$ from $\mathcal{T}$ to simulate the cold-start setting. For each item in $\mathcal{T}^c$, which corresponds to a task, we preserve m examples for both the support set and the query set (a total of $2m$ examples). The remaining examples of these items are dropped since they should not appear before we train the RM-IdEG. To avoid the performance of the base model being decreased, we limit each item in $\mathcal{T}^c$ to having less than or equal to M examples $(M > 2m)$ and obviously, greater or equal to $2m$ examples. We denote the constructed cold-start dataset by $\mathcal{D}^c$. The data examples of the remaining items $\mathcal{T} - \mathcal{T}^c$ constitute the warm-start dataset $\mathcal{D}^w$. Remark that the items in $\mathcal{T}$ are all warm-start items since they have at least one data example. We call $\mathcal{D}^c$ cold-start because they are used to train RM-IdEG, which is designed for totally cold-start items. Also, $\mathcal{D}^w$ is called warm-start since it is used to train DisNet, which does not consider the cold-start problem.

In summary, we have three datasets: (1) an auxiliary dataset $\mathcal{D}^a$, having no scenario-specific contexts, collected from other domains; (2) a warm-start dataset $\mathcal{D}^w$ that has rich contextual information; and (3) a cold-start dataset $\mathcal{D}^c$ that contains few-shot examples. Accordingly, the whole model is trained in three stages, and we put the details in Algorithm 1.

4. Experiments

To justify the effectiveness of DisNet and RM-IdEG, we conduct comprehensive experiments to answer the following questions:

RQ1: can DisNet outperform state-of-the-art methods?

RQ2: can RM-IdEG outperform state-of-the-art IdEGs?

RQ3: is our model sensitive to the parameters?

```
Input: $\mathscr{D}^a$: auxiliary dataset
Input: $\mathscr{D}^w$: warm-start dataset
Input: $\mathscr{D}^c$: cold-start dataset
Input: $(\tilde{t}, \tilde{u}, \mathbf{c}, \mathbf{c}_s)$: a testing example
Output: $\tilde{y}^{\mathrm{DN}}$: the predicted label of $(\tilde{t}, \tilde{u}, \mathbf{c}, \mathbf{c}_s)$
repeat
    /* the first stage, pretrain the model using auxiliary data */
    Randomly sample a batch of data from $\mathscr{D}^a$
    Calculate the predicted label $\hat{y}$ by equation (1)
    Update $g$, $f_u$, $f_t$ by gradient descent
until Converge
Fix $g$, $f_u$, $f_t$ to $\overline{g}$, $\overline{f}_u$, $\overline{f}_t$
repeat
    /* the second stage, train DisNet using warm-start data */
    Randomly sample a batch of data from $\mathscr{D}^w$
    Calculate $\mathbf{q}_u$, $\mathbf{p}_t$ using $\overline{f}_u$, $\overline{f}_t$
    Compute the shifted interest vector $\mathbf{q}_u^s$ by equation (6)
    Calculate the predicted label $\hat{y}^{\mathrm{DN}}$ using $\overline{g}$ by equation (7)
    Update $h$, $\mathrm{ISO}(\cdot)$ by gradient descent
until Converge
Fix all the trainable parameters except the item id embeddings
repeat
    /* the third stage, train RM-IdEG using cold-start data */
    Randomly sample an item $t_i$ and get its support/query sets $(\mathscr{D}_i^s, \mathscr{D}_i^q)$ from $\mathscr{D}^c$
    Aggregate embeddings of relational items of $t_i$ by equation (11)
    Generate an id embedding $\mathbf{e}_t^*$ for $t_i$ using RM-IdEG
    Compute the cold-start loss on $\mathscr{D}_i^s$ by equation (2)
    Update the id embedding of $t_i$ to $\mathbf{e}_t'$ by equation (3)
    Compute the warmed loss on $\mathscr{D}_i^q$ by equation (4)
    Update RM-IdEG by gradient descent
until Converge
if $\tilde{t}$ is a cold-start item then
    Generate an id embedding $\tilde{\mathbf{e}}_t$ for $\tilde{t}$ using RM-IdEG
else
    Get the id embedding $\tilde{\mathbf{e}}_t$ of $\tilde{t}$ from the embedding layer
end if
Return a label $\tilde{y}^{\mathrm{DN}}$ for $(\tilde{t}, \tilde{u}, \mathbf{c}, \mathbf{c}_s)$ by equation (7) using DisNet
```

Algorithm 1: The training and testing procedure.

4.1. Dataset

4.1.1. Dataset Description. We evaluate our methods on two synthetic datasets and a real-world dataset:

MovieLens (https://grouplens.org/datasets/movielens/) [38]: it consists of 1.0 million movie-ranking instances across about 6,000 users and 4,000 movies. The features of movies include movie id, title, year of release, and genres. Titles and genres are lists of tokens. The features of users include user id, age, gender, occupation, and zipcode. To simulate our fresh item setting, we choose gender, occupation, and zipcode as scenario-specific context features. We also convert the rating scores to binary values. The ratings smaller than 4 are turned into 0, and the others are turned into 0.

Book-Crossing (http://www2.informatik.uni-freiburg.de/cziegler/BX/) [39]: it is collected by Cai-Nicolas Ziegler in a one-month crawl from the Book-Crossing (http://www.bookcrossing.com/) community. It contains 0.27 million users, providing 1.15 million ratings about 0.28 million books. The features of books include ISBN number (book id), book title, year of publication, and publisher. The features of users include age and location. Similar to MovieLens, we select location as a scenario-specific context feature. The ratings are converted to 1 if they are at least 4 and 0, otherwise.

Taobao-Fresh: it collects 203.1 million user-item click interactions produced by the main entrance page of Taobao's app as auxiliary data and 4.4 million user-item click interactions produced by the *New Tendency* page as fresh-item recommendation data. A total of 4.8 million users and 1.6 million items are considered, with 71 user features, 17 item features, and 17 contextual features (auxiliary data have no contexts).

4.1.2. Data Splitting. For MovieLens and Book-Crossing, we first group the items by their ids. We put those items with the number of examples less than $M + 1$ and larger than $2m - 1$ in $\mathscr{T}^c$. Then, we construct a cold-start dataset $\mathscr{D}^c$ by preserving $2m$ examples for each item. From the examples of the remaining items $\mathcal{T} - \mathcal{T}^c$, we randomly choose 80% as auxiliary data $\mathcal{D}^a$ and 20% as the warm-start dataset $\mathcal{D}^w$. We set $m = 20$ and $M = 100$ for MovieLens. For Book-Crossing, we notice that a total of 48,434 books are rated by exactly 2 users. Hence, we set $m = 1$ and $M = 2$. It enables us to study an extreme experimental setting, i.e., each cold-start item is one-shot.

For Taobao-Fresh, the auxiliary data $\mathcal{D}^a$ have been collected. We then split the fresh-item recommendation data into two parts. The first one is a cold-start dataset $\mathcal{D}^c$ where items have greater than or equal to 10 interactions and less than or equal to 20 interactions. Similarly, each item in $\mathcal{D}^c$ has a support set and a query set, each of which has 5 examples. The examples of the remaining items are collected as the warm-start dataset $\mathcal{D}^w$. The statistics of these datasets can be found in Table 2.

4.1.3. Data Generation. To answer RQ1, for each dataset, we run DisNet on three types of data:

Auxiliary-only data: they contain the auxiliary data and context-free warm-start data, i.e., the context features of the warm-start data are removed.

Context-only: it is exactly warm-start data. In other words, DisNet is run without pretraining.

Full data: they comprise auxiliary data and warm-start data and are the main setting of this paper.

Note that the three types of data are used to test the effectiveness of DisNet, while cold-start data are used to evaluate the superiority of RM-IdEG.

For performance evaluation, we randomly divide the warm-start and cold-start data into 80% training and 20% testing. We run the experiments for five times, and the mean AUC performance on the testing set is reported.

4.2. Baselines. We evaluate the proposed model in two stages. In the first stage, we compare DisNet with three context-aware recommendation models:

DeepFM [11]: it feeds embeddings to a factorization machine model as well as a multilayer perceptron and then aggregates their outputs and gets the final prediction.

PNN [13]: the dense embeddings are fed into a dense layer and a product layer. Then, it concatenates their outputs together and uses a two-layer neural network to get the prediction.

CFM [15]: CFM is a recent state-of-the-art CARS method that explicitly learns second-order feature interactions. It calculates the pairwise outer product of dense embeddings and stacks them to obtain an interaction cube. Then, it applies the convolution pooling technique to get the final prediction.

The dimension of embedding vectors of each input field is fixed to 128, and the activation function is chosen as ReLU for all the models. As suggested in [11], we use three dense hidden layers as the deep component for both DeepFM and PNN. For DisNet, the size of the user/item latent representation is set as 64. We use two fully connected layers with a hidden dimension of 64 for user/item representation networks as well as the decision-making network. We do not activate the outputs of user/item representation networks. The context network of the NN/add ISO and the shifting network of the NN ISO also comprise two fully connected layers with hidden size 64 and without activation in the final layer. For the COT ISO, we linearly learn a contextual operation matrix of size 64×64 from the contexts. Finally, the learning rate and l_2-regularization parameters are fine-tuned by five-fold cross-validation.

Then, we evaluate the RM-IdEG with two baselines:

Rand-IdEG: the random initialization of id embeddings is one of the most commonly used strategies in recommender systems.

Meta-IdEG [27]: the state-of-the-art solution to the cold-start problem. It firstly feeds the item features into a simple neural network to generate embeddings and then trains them in a learning-to-learn manner.

For Rand-IdEG, we initialize the id embeddings with random values from a standard Gaussian distribution with standard deviation 0.01. For meta-IdEG, we use the neural network architecture as suggested in [27]. For RM-IdEG, we use a two-layer neural network with a hidden size of 128 as the IdEG network. According to Pan et al. [27], the tradeoff parameter η is robust. Hence, we follow their experimental setting and set η as 0.1 for meta-IdEG and RM-IdEG. We also follow their two suggestions that use tanh as activation and remove the bias of the output layer. For a target item in the synthetic dataset, we choose k-nearest neighbors from the previous training dataset, i.e., $\mathcal{D}^w \cup \mathcal{D}^a$, using hamming distance as the relevant items, where k is chosen by five-fold cross-validation. For Taobao-Fresh, we randomly select 10 items having the same seller and 10 items having the same brand as the relevant items. We choose DisNet-NN as the base model, which has been pretrained by $\mathcal{D}^a$ and $\mathcal{D}^w$.

4.3. Empirical Results

4.3.1. Performance Comparison of Context-Aware Models (RQ1). Tables 3 and 4 report the testing AUC comparison of three context-aware models on two synthetic datasets and the Taobao-Fresh dataset. We have the following findings:

All the methods obtain the best performance on the full data. For example, on Taobao-Fresh, DisNet-NN improves the AUC scores on auxiliary-only and context-only data by 1.00% and 1.69%, respectively. This finding verifies the importance of utilizing auxiliary data and contexts to alleviate the data-deficiency problem.

On the Taobao-Fresh dataset, all the methods achieve significantly greater improvement on the context-only data than the auxiliary-only data. It demonstrates that, in the fresh-item recommendation task, the context information highly reflects the user interest.

Table 2: The statistics of the datasets.

Datasets	# users	# items	# interactions (M)†	# auxiliary (M)	# warm-start (M)	# cold-start‡
Book-Crossing	0.28 M	0.27 M	1.15	0.75	0.19	48,434
MovieLens	6,040	3,706	1.00	0.76	0.19	30,645
Taobao-Fresh	4.78 M	1.61 M	207.54	203.13	4.09	0.31 M

†The first three columns list the number of users, items, and data examples of the whole dataset. ‡The second three columns report the number of data examples of the preprocessed datasets after splitting.

Table 3: Testing AUC comparison of context-aware models on synthetic datasets.

Methods	Dataset: Book-Crossing			Dataset: MovieLens		
	Auxiliary†	Context	Full data	Auxiliary†	Context	Full data
DeepFM	0.7836●	0.7741●	0.7840●	0.7284●	0.7586●	0.7654●
PNN	0.7854●	0.7725●	0.7857●	0.7290●	0.7584●	0.7649●
CFM	0.7730●	0.7726●	0.7707●	0.7289●	0.7571●	0.7633●
DisNet-Add	–	0.7745°	0.7862●	–	0.7576°	0.7660●
DisNet-COT	–	0.7733●	0.7864●	–	0.7559●	0.7664°
DisNet-NN	0.7858	0.7748	0.7878	0.7287	0.7577	0.7666

●/° indicates whether the DisNet-NN variant is significantly superior to the coupling algorithm or not on each dataset (pairwise t-test at the 0.05 significance level).†On the auxiliary-only data, the network architecture of DisNet is fixed, and we only report the performance once.

On auxiliary-only data, all the models are competitive with each other. However, on full data, the performance of baselines shows no significant improvement after the context features being involved. The reason is these baselines deeply couple the context in the model, and thus, the knowledge of the auxiliary domain cannot be fully utilized. Take DeepFM as an example; since $\mathscr{D}^a$ and $\mathscr{D}^w$ have different input formats, the deep component cannot be reused. Though we can reuse the embedding layer, its predictive performance is limited.

DisNet models with full data significantly outperform all the baselines as well as their auxiliary-only and context-only counterparts. The interest-shifting operator enables us to completely exploit both context and cross-domain information.

Different interest-shifting operators show competitive performance with each other. Moreover, the NN-based operator obtains the best performance because it enables the user interest to be shifted nonlinearly.

Interestingly, DisNet-COT always underperforms DisNet-Add on the context-only dataset but is better than DisNet-Add on the full dataset. We suppose the reason is the COT operator tends to overfit on context-only data since it contains more parameters. With the help of auxiliary data, this problem is alleviated.

4.3.2. Performance Comparison of Different IdEGs (RQ2). Tables 5 and 6 list the cold-start and warmed-up performance of DisNet with different id embedding generators. Once the IdEG produces the id embeddings, the cold-start performance is directly evaluated on a meta-testing query set, where all items are cold-start ones. Then, we perform one step of gradient descent to update the id embeddings using a meta-testing support set that contains the same items as the query set. Finally, the warmed-up performance is evaluated again on the query set.

Table 4: Testing AUC comparison of context-aware models on the Taobao-Fresh dataset.

Methods	Auxiliary†	Context	Full data
DeepFM	0.7367●	0.7449●	0.7362●
PNN	0.7413●	0.7439●	0.7417●
CFM	0.7377●	0.7441●	0.7442●
DisNet-Add	–	0.7480°	0.7528●
DisNet-COT	–	0.7467●	0.7533°
DisNet-NN	0.7409	0.7483	0.7534

●/° indicates whether the RM-IdEG variant is significantly superior to the coupling algorithm or not (pairwise t-test at the 0.05 significance level).†On the auxiliary-only data, the network architecture of DisNet is fixed, and we only report the performance once.

From the results, we conclude that

Meta-IdEG and RM-IdEG outperform Rand-IdEG on both cold-start and warmed-up phases because the learning-to-learn training procedure guarantees them to quickly achieve good generalization ability on unseen data.

RM-IdEG achieves the best performance on all the datasets. In particular, even with one-shot training, RM-IdEG still outperforms on the Book-Crossing dataset. By integrating information of significantly relevant items, RM-IdEG inherently models the community structural information when initializing id embeddings.

4.3.3. Parameter Sensitivities (RQ3). The main parameters are the tradeoff parameter of the meta-loss η and the number of relevant items k. The robustness of η has been studied in [27]. Thus, we investigate the sensitivity of k and the results on Book-Crossing and MovieLens datasets which are shown in Figure 3. We can see that when k is small, the performance is close to Meta-IdEG because few relational information is learned. The best result is obtained when $k = 6$, and then the

Table 5: Testing AUC comparison of different IdEGs on synthetic datasets.

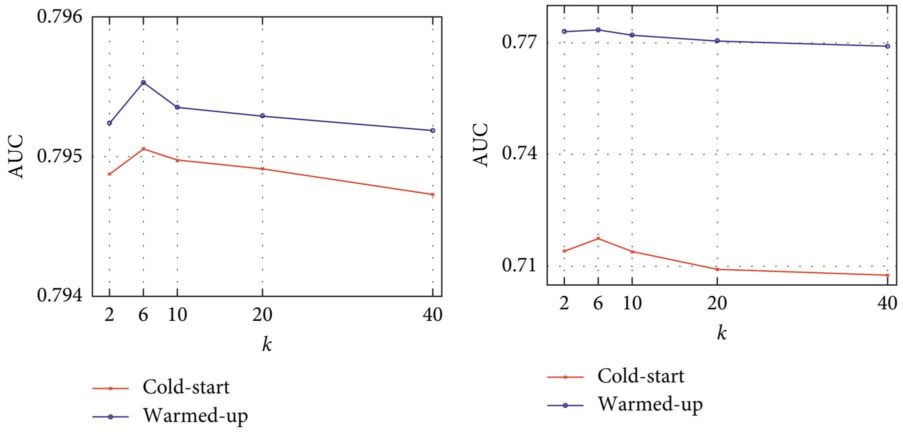
IdEG types	Dataset: Book-Crossing		Dataset: MovieLens	
	Cold-start	Warmed-up	Cold-start	Warmed-up
Rand-IdEG	0.7940 (+0.00%)•	0.7943 (+0.00%)•	0.7065 (+0.00%)•	0.7377 (+0.00%)•
Meta-IdEG	0.7945 (+0.06%)•	0.7948 (+0.06%)•	0.7132 (+0.95%)•	0.7680 (+4.11%)•
RM-IdEG	0.7951 (+0.14%)	0.7955 (+0.15%)	0.7174 (+1.54%)	0.7735 (+4.85%)

•/° indicates whether the DisNet-NN variant is significantly superior to the coupling algorithm or not on each dataset (pairwise *t*-test at the 0.05 significance level).

Table 6: Testing AUC comparison of different IdEGs on the Taobao-Fresh dataset. The best ones are shown in bold.

IdEG types	Cold-start	Warmed-up
Rand-IdEG	0.5792 (+0.00%)•	0.6042 (+0.00%)•
Meta-IdEG	0.6133 (+5.89%)•	0.6361 (+5.28%)•
RM-IdEG	**0.6160 (+6.35%)**	**0.6382 (+5.63%)**

•/° indicates whether RM-IdEG is significantly superior to the coupling algorithm or not (pairwise *t*-test at the 0.05 significance level).



(a)

(b)

Figure 3: Parameter sensitivity of k on two datasets. (a) Book-Crossing. (b) MovieLens.

performance drops. The reason is that, as k becomes larger, the relations become weaker, but the model complexity increases.

5. Related Work

5.1. Context-Aware Recommendation. Context-aware recommender systems (CARSs) have attracted considerable attention in past years [7]. Early work in CARS can be divided into two categories: (1) prefiltering methods [40], where context guides the selection of training data; (2) postfiltering methods [41], where context drives recommendation results' selection. The main limitation of these methods is that they require the supervision and fine-tuning in all steps of recommendation [42]. To address this problem, contextual modeling approaches capture the contextual information directly in model construction. Some works are based on matrix factorization [8], such as CAMF [28] and CSLIM [9]. Another group of studies exploits tensor factorization techniques for modeling user-item-context relations [43, 44]. Recently, factorization machines [42, 45, 46] and deep learning [47, 48] based on CARS become increasingly popular, which directly model nonlinear interactions between features. Some studies also use representation learning techniques, e.g., CARS^2 [49] and COT [37], which provide not only a latent vector but also context-aware representations. In summary, all the above methods assume the data are sufficient for training, while severe data-deficiency problem occurs in many fresh-item recommendation pages.

5.2. Cross-Domain Recommendation. As we have discussed, data deficiency is one of the most challenging problems for recommender systems, and it is much more significant in many fresh-item recommendation scenarios. One promising solution to this problem is cross-domain recommender systems (CDRSs) [50]. Existing CDRSs can be categorized into symmetric and asymmetric ones. Symmetric models [16, 18, 51, 52] collect sparse data from multiple domains and anticipate that these domains can complement each other. In our task, symmetric strategy is incompatible because the two domains have heterogeneous data format and imbalance data size. Thus, we consider asymmetric models [19, 20, 21], which aim to leverage data in an auxiliary domain to alleviate data deficiency of the target domain. In this way, knowledge learned from the auxiliary domain is directly transferred to the target domain, acting as priors or regularization. Nevertheless, many asymmetric CDRSs adopt shallow methods and have the difficulty in learning complex user-item interaction relationship [18, 26]. Moreover, scenario-specific contextual information of the target domain has been seldom considered.

5.3. Cold-Start Recommendation. When recommending cold-start fresh items, a severe cold-start problem occurs. To handle this problem, it is common to collect information for the cold item or user, e.g., item attributes [22, 23] and user attributes [24, 25]. A recent work HERS [26] also utilizes relational data to boost performance, such as social information of users. In [16], the authors explored a symmetric cross-domain recommender system, where shared knowledge can help alleviate the cold-start problem.

Recently, a series of works [27, 53, 54] also adopt meta-learning technique [55] which enables the recommender system to achieve good generalization ability after few-shot training. From the cold-start user perspective, MeLU [53] learns a meta-id embedding for the cold-start users and then predicts the user preference on the items by the norm of gradients. From the cold-start item perspective, Pan et al. [27] proposed the meta-Id embedding generator (meta-IdEG), which also takes id embedding initialization into account. However, since meta-IdEG only uses item features to generate id embedding, it ignores the community structural information concealed in id embedding, which leads to a suboptimal solution.

6. Discussion and Conclusion

6.1. Further Discussion. In this section, we discuss the significance of this work.

6.1.1. Importance of the Application. The fresh-item recommendation task reveals a new perspective of personalized recommendation, i.e., the impact of items' life period. Some people may prefer products which stand the test of time, while some others may be interested in newly released products. The *New Tendency* page enables the latter ones' recommendation to be fully personalized. From another point of view, these fresh items also obtain more opportunities to be exposed. Hence, high-quality and novel products can quickly become popular. We also address the main difficulties of this learning task, i.e., data deficiency and cold-start.

6.1.2. Importance of the Techniques. Surprisingly, though the two techniques DisNet and RM-IdEG are proposed to handle the fresh-item recommendation task, we find that both methods have a wide range of applications.

As aforementioned, the DisNet is designed for fresh-item recommendation pages. Actually, such pages are quite common in existing E-commerce platforms. For example, after a bill being paid, the E-commerce platform will recommend other related items to the customers. It is a classical fresh-item recommendation scenario. Obviously, a fresh-item recommendation page usually contains rich contextual information. The contexts reflect that the user interest shifts from a general one to a scenario-specific one. However, with fewer page views, such pages usually face severe data-deficiency problems. And this work can address this issue by giving a novel learning framework, which simultaneously transfers knowledge from an auxiliary domain as well as fully utilizes the context information.

RM-IdEG can also be applied to many real-world applications. In [27], the authors proposed to learn meta-id-embeddings for cold-start advertisements. And we can also collect relevant advertisements by its company, topic, and so on. As a result, the model can generate better id embeddings. Furthermore, other relational data can also be considered. For instance, if we consider the user cold-start problem [53], we may explore the social networks of a new user so that RM-IdEG is able to initialize a fast-adapting and relation-aware id embedding.

6.2. Conclusion. In this work, we address two difficulties of the fresh-item recommendation task. First, we propose a deep interest-shifting network to deal with the data-deficiency problem of fresh item recommendation. Specifically, users' general interests are learned from a huge number of an auxiliary dataset. Then, our model shifts the user interest to a scenario-specific one using context features. Second, we propose a relational meta-Id-embedding generator (RM-IdEG) to alleviate the cold-start problem. RM-IdEG is trained in a learning-to-learn manner with relational information being integrated. Hence, community structural information can be inherently embedded in the id embeddings of newborn items. Extensive experiments on two synthetic datasets and a real-world dataset clearly identify the effectiveness of our approaches, which have been already deployed on a large-scale online fresh-item recommendation application.

Data Availability

Previous reported data were used to support this study, and these prior studies (and datasets) are cited at relevant places within the text as references [38, 39].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61972336).

References

[1] J. Song, Li Zhao, Z. Hu et al., "Poisonrec: an adaptive data poisoning framework for attacking black-box recommender systems," in *Proceedings of the ICDE 2020*, pp. 157–168, Dallas, TX, USA, 2020.

[2] P. Wang, Li Zhao, X. Pan, D. Ding, X. Chen, and Y. Hou, "Density matrix based preference evolution networks for e-commerce recommendation," in *DASFAA*, pp. 366–383, Springer, Berlin, Germany, 2019.

[3] P. Wang, Li Zhao, Y. Zhang, Y. Hou, and L. Ge, "QPIN: a quantum-inspired preference interactive network for E-commerce recommendation," in *Proceedings of the CIKM*, pp. 2329–2332, Beijing, China, 2019.

[4] J. Chen, Z. Wang, T. Zhu, and F. E. Rosas, "Recommendation algorithm in double-layer network based on vector dynamic evolution clustering and attention mechanism," *Complexity*, vol. 2020, no. 2020, 19 pages, Article ID 5206087.

[5] Z. Li, F. Xiong, X. Wang, H. Chen, and Xi Xiong, "Topological influence-aware recommendation on social networks," *Complexity*, vol. 2019, Article ID 6325654, 12 pages, 2019.

[6] Li Zhao, L. Zhang, C. Lei, X. Chen, J. Gao, and J. Gao, "Attention with long-term interval-based deep sequential learning for recommendation," *Complexity*, vol. 2020, Article ID 6136095, 13 pages, 2020.

[7] G. Adomavicius, B. Mobasher, F. Ricci, and T. Alexander, *Context-Aware Recommender Systems*, AI Magazine, Palo Alto, CA, USA, 2011.

[8] M. H. Abdi, O. George, and R. W. Mwangi, "Matrix factorization techniques for context-aware collaborative filtering recommender systems: a survey." *Computer and Information Science*, vol. 11, no. 2, 2018.

[9] Y. Zheng, B. Mobasher, D Robin, and Burke, "Deviation-based contextual SLIM recommenders," in *Proceedings of the CIKM*, pp. 271–280, Shanghai, China, November 2014.

[10] S. Wang, J. Zhang, and X.D. Zhang, "A time-aware CNN-based personalized recommender system," *Complexity*, vol. 2019, Article ID 9476981, 11 pages, 2019.

[11] 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 IJCAI 2017*, pp. 1725–1731, Melbourne, Australia, 2017.

[12] L. Guo, H. Jiang, X. Liu, and C. Xing, "Network embedding-aware point-of-interest recommendation in location-based social networks," *Complexity*, vol. 2019, Article ID 3574194, 18 pages, 2019.

[13] Y. Qu, B. Fang, W. Zhang et al., "Product-based neural networks for user response prediction over multi-field categorical data," *ACM Transactions on Information Systems*, vol. 37, no. 1, 2019.

[14] O. Tal and Y. Liu, "A joint deep recommendation framework for location-based social networks," *Complexity*, vol. 2019, Article ID 2926749, 11 pages, 2019.

[15] X. Xin, Bo Chen, X. He, D. Wang, Y. Ding, and J. Jose, "CFM: convolutional factorization machines for context-aware recommendation," in *Proceeding of the IJCAI*, pp. 3926–3932, Macao, China, 2019.

[16] Da Cao, X. He, L. Nie et al., "Cross-platform app recommendation by jointly modeling ratings and texts," *ACM Transactions on Information Systems*, vol. 35, no. 4, 2017.

[17] C. Gao, X. Chen, F. Feng et al., "Cross-domain recommendation without sharing user-relevant data," 2019.

[18] G. Hu, Yu Zhang, and Q. Yang, "CoNet: collaborative cross networks for cross-domain recommendation," in *Proceedings of the CIKM 2018*, pp. 667–676, Turin, Italy, 2018.

[19] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," 2013.

[20] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction," in *Proceedings of the IJCAI*, pp. 2052–2057, Pasadena, CA, USA, 2009.

[21] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in *Proceedings of the AAAI*, Atlanta, GA, USA, 2010.

[22] S. Martin and M. Amin, "Item cold-start recommendations: learning local collective embeddings," in *Proceedings of the RecSys*, pp. 89–96, Foster City, CA, USA, 2014.

[23] M. Volkovs, G. W. Yu, and T. Poutanen, "DropoutNet: addressing cold start in recommender systems," 2017.

[24] S. Roy and S. C. Guntuku, "Latent factor representations for cold-start video recommendation," in *Proceedings of the RecSys 2016*, pp. 99–106, Boston, MA, USA, 2016.

[25] Y. Seroussi, F. Bohnert, and I. Zukerman, "Personalised rating prediction for new users using latent factor models," 2011.

[26] L. Hu, S. Jian, L. Cao, Z. Gu, Q. Chen, and A. Amirbekyan, "HERS: modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3830–3837, 2019.

[27] F. Pan, S. Li, Ao Xiang, P. Tang, and Q. He, "Warm up cold-start advertisements: improving CTR predictions via learning to learn ID embeddings," , pp. 695–704, 2019.

[28] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the RecSys 2011*, pp. 301–304, Chicago, IL, USA, October 2011.

[29] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: deep knowledge-aware network for news recommendation," 2018.

[30] Q. Zhu, X. Zhou, Z. Song, J. Tan, and L. Guo, "DAN: deep attention neural network for news recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5973–5980, 2019.

[31] F. Liu, S. Xue, J. Wu et al., "Deep learning for community detection: progress, challenges and opportunities," in *Proceedings of the IJCAI*, pp. 4981–4987, Yokohama, Japan, 2020.

[32] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 416–429, 2014.

[33] J. Wu, X. Zhu, C. Zhang, and S. Y. Philip, "Bag constrained structure pattern mining for multi-graph classification," *Ieee Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2382–2396, 2014.

[34] X. Chen, G. Yu, J. Wang, C. Domeniconi, Li Zhao, and X. Zhang, "ActiveHNE: active heterogeneous network embedding," in *Proceedings of the IJCAI 2019*, pp. 2123–2129, Macao, China, 2019.

[35] C. Chu, Li Zhao, B. Xin et al., "Deep graph embedding for ranking optimization in e-commerce," in *Proceedings of the CIKM*, pp. 2007–2015, Turin, Italy, 2018.

[36] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI*, pp. 2181–2187, Austin, TX, USA, 2015.

[37] S. Wu, Q. Liu, L. Wang, and T. Tan, "Contextual operation for recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2000–2012, 2016.

[38] F. Maxwell Harper and J. A. Konstan, "The movielens datasets: history and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, p. 19, 2016.

[39] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," 2005.

[40] G. Adomavicius, S. Ramesh, S. Sen, and T. Alexander, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23, p. 1, 2005.

[41] U. Panniello, T. Alexander, M. Gorgoglione, C. Palmisano, and A. Pedone, "Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems," in *Proceedings of the RecSys 2009*, pp. 265–268, Copenhagen, Denmark, 2009.

[42] W.-Y. Ma, J.-Y. Nie, R. Baeza-Yates, T.-S. Chua, and W. Bruce Croft, 2011, SIGIR: ACM.

[43] B. Hidasi and T. Domonkos, *Fast ALS-Based Tensor Factorization for Context-Aware Recommendation from Implicit Feedback*, pp. 67–82, Springer, Berlin, Germany, 2012.

[44] Y. Shi, A. Karatzoglou, L. Baltrunas, M. A. Larson, A. Hanjalic, and N. Oliver, "TFMAP: optimizing map for top-n context-aware recommendation," 2012.

[45] X. Chen, Y. Zheng, J. Wang, W. Ma, and J. Huang, "RaFM: rank-aware factorization machines," in *Proceedings of the ICML 2019*, pp. 1132–1140, Long Beach, CA, USA, 2019.

[46] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Top-N recommendation with multi-channel positive feedback using factorization machines," *ACM Transactions on Information Systems*, vol. 37, no. 2, 2019.

[47] H.-T. Cheng, L. Koc, J. Harmsen et al., "Wide & deep learning for recommender systems," in *Proceedings of the RecSys 2016*, pp. 7–10, Boston, MA, USA, 2016.

[48] G. Zhou, X. Zhu, C. Song et al., "Deep interest network for click-through rate prediction," in *Proceedings of the KDD*, pp. 1059–1068, London, UK, 2018.

[49] Y. Shi, A. Karatzoglou, L. Baltrunas, M. A. Larson, and A. Hanjalic, "CARS2: learning context-aware representations for context-aware recommendations," in *Proceedings of the CIKM*, pp. 291–300, Shanghai, China, 2014.

[50] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci, "Cross-domain recommender systems: a survey of the state of the art," in *Proceedings of the Spanish Conference on Information Retrieval*, Valencia, Spain, 2012.

[51] C.-Y. Li and S.-D. Lin, "Matching users and items across domains to improve the recommendation quality," in *Proceedings of the KDD 2014*, pp. 801–810, New York, NY, USA, 2014.

[52] M. Tong, H. Shen, X. Jin, and X. Cheng, "Cross-domain recommendation: an embedding and mapping approach," in *Proceedings of the IJCAI 2017*, pp. 2464–2470, Melbourne, Australia, 2017.

[53] H. Lee, Im Jinbae, S. Jang, H. Cho, and S. Chung, "MeLU: meta-learned user preference estimator for cold-start recommendation," in *Proceedings of the KDD 2019*, pp. 1073–1082, Anchorage, AK, USA, 2019.

[54] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle, "A meta-learning perspective on cold-start recommendations for items," 2017.

[55] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the ICML*, pp. 1126–1135, Sydney, Australia, August 2017.