Inter- and Intra-Domain Relation-Aware Heterogeneous Graph Convolutional Networks for Cross-Domain Recommendation

Ke Wang[1,2], Yanmin Zhu[1,2](✉), Haobing Liu[1], Tianzi Zang[1,2], Chunyang Wang[1,2], and Kuan Liu[1,2]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
{onecall,yzhu,liuhaobing,zangtianzi,wangchy,lk_shjd}@sjtu.edu.cn
[2] State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, MNR, Hangzhou, China

Abstract. Cross-domain recommendation aims to model representations of users and items with the incorporation of additional knowledge from other domains, so as to alleviate the data sparsity issue. While recent studies demonstrate the effectiveness of cross-domain recommendation systems, there exist two unsolved challenges: (1) existing methods focus on transferring knowledge to generate shared factors implicitly, which fail to distill domain-shared features from explicit cross-domain correlations; (2) The majority of solutions are unable to effectively fuse domain-shared and domain-specific features. To this end, we propose **I**nter- and **I**ntra-domain **R**elation-aware **C**ross-**D**omain **R**ecommendation framework (I^2RCDR) to explicitly learn domain-shared representations by capturing high-order inter-domain relations. Specifically, we first construct a cross-domain heterogeneous graph and two single-domain heterogeneous graphs from ratings and reviews to preserve inter- and intra-domain relations. Then, a relation-aware graph convolutional network is designed to simultaneously distill domain-shared and domain-specific features, by exploring the multi-hop heterogeneous connections across different graphs. Moreover, we introduce a gating fusion mechanism to combine domain-shared and domain-specific features to achieve dual-target recommendation. Experimental results on public datasets show that the effectiveness of the proposed framework against many strong state-of-the-art methods.

Keywords: Cross-domain recommendation · Inter-domain relations · Relation-aware graph convolutional network · Gating mechanism

1 Introduction

Recommendation systems learn representations from interactions between users and items. Many traditional methods [1,11] leverage historical feedbacks (e.g.,

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13246, pp. 53–68, 2022.
https://doi.org/10.1007/978-3-031-00126-0_4

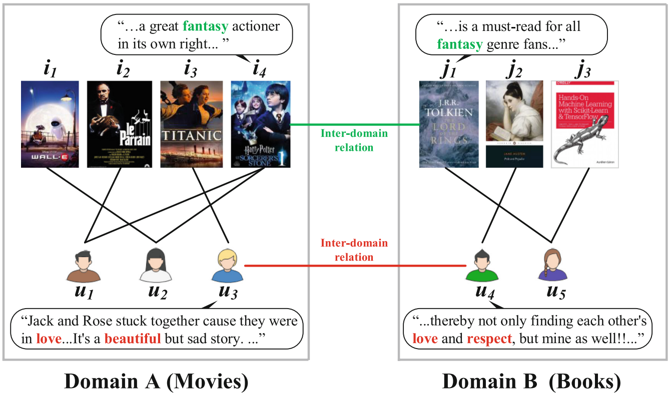


Fig. 1. An illustrative example of inter-domain relations for CDR systems. Both the reviews of items i_4 and j_1 are about the fantasy genre, while both the user u_3 and u_4 commented on the love theme in their reviews.

ratings) to capture users' preferences on items. However, the user-item interactions are usually sparse, which makes recommendation systems unable to generate optimal representations of users who give few ratings. Therefore, recent efforts [3,17,31] merge extra information from auxiliary domains into target domains to build Cross-Domain Recommendation (CDR).

Existing CDR approaches can be divided into two categories, transferring knowledge in different directions. The first category transfers in a unidirectional way, which focuses on learning the features of users and items from an auxiliary domain and then transferring learned features to a target domain. The most popular unidirectional technique is EMCDR [20], which learns a mapping function to execute target-domain recommendation utilizing the two-stage embedding-and-mapping paradigm [7,32]. However, the unidirectional method could easily accumulate noise in the intermediate steps and fall into sub-optimal learning [30]. The second category transfers in a bidirectional way. Considering that each domain is considerably richer in particular sorts of information, some recent works leverage bidirectional knowledge transfer to improve the performance on both domains simultaneously. This category selects overlapped users (or items) as a bridge to extract shared factors or patterns, which are transferred between two domains to achieve dual-target CDR [12,16,29].

Unfortunately, the aforementioned methods implicitly transfer knowledge to generate shared factors, which fail to distill domain-shared features from explicit correlations between cross-domain users (or items). In CDR systems, there are many inter-domain user-user and item-item relations, which are domain independent and could be used to explicitly portray domain-shared features. For example, as shown in Fig. 1, the reviews of movie i_4 in domain A and book j_1 in domain B which contain similar content (highlighted in boldface) show that the two items both belong to the fantasy genre, so we think the two items have the same property and there exists the inter-domain item-item relation. Similarly, the users u_3 and u_4 have the same preference and there exists the inter-domain user-user relation. We argue that such inter-domain relations are beneficial for learning domain-shared features, which are ignored by existing approaches.

Moreover, most models are incapable of fusing domain-shared and domain-specific features efficiently. In fact, user's (or item's) features are comprised of domain-shared part and domain-specific part, which reveal the cross-domain consistency and single-domain peculiarity, respectively [15]. Recently, emerged models simply conduct combination operations over domain-shared and domain-specific features, such as concatenation or addition [5,17]. However, the linear integration makes these models unable to fully explore the nonlinear interaction between the two kinds of features.

In this paper, we aim to capture inter-domain relations associating different domains to explicitly learn domain-shared features of users and items. However, there are a few challenges. The first challenge is how to explicitly establish the inter-domain relations between users (or items) from different domains. The most of existing CDR methods only model the cross-domain interactions from distinct domains, which are insufficient. Data from different domains may have similar semantic relations. Book and movie as an example, contents on both domains have some common topics, e.g., genre, plot, scene. As such, a good CDR model should have the ability to capture correlated information across domains. The second challenge is how to respectively distill domain-shared and domain-specific features. Capturing inter-domain relations results in a reconsideration for the current CDR methods that implicitly model intra-domain relations to generate domain-shared features. We should distinguish inter-domain and intra-domain relations to separately distill domain-shared and domain-specific features. Third, an effective strategy should be used to fuse domain-shared and domain-specific features, aiming to adaptively assign weights to balance cross-domain consistency and single-domain peculiarity.

To tackle these challenges, we propose a graph-structured CDR framework, namely I^2RCDR, which can collectively model the high-order **I**nter- and **I**ntra-domain **R**elations for dual-target **C**ross-**D**omain **R**ecommendation. First, we construct a cross-domain heterogeneous graph from interactions (ratings and reviews) to preserve inter-domain relations (including user-user and item-item relations) and user-item relations within each domain. Motivated by the landmark research BERT [6], we apply the SentenceBERT [21], which fine-tunes the BERT model and can transform reviews into semantic embeddings, to model the inter-domain relations. Meanwhile, a single-domain heterogeneous graph for each domain is also designed to represent intra-domain relations (including user-user and item-item relations) and user-item relation. Second, we introduce a relation-aware graph convolutional network (GCN) to extract domain-shared and domain-specific features, which can not only explore the multi-hop heterogeneous connections between users and items but also inject relation property into embedding propagation by encoding multi-typed relations. Finally, to achieve dual-target recommendation, we propose a gating fusion mechanism that can share partial parameters to seamlessly combine domain-shared and domain-specific features. The main contributions of this paper are as follows:

- We explore inter-domain relations to learn domain-shared representations of users and items explicitly. Three heterogeneous graphs are constructed from ratings and reviews to preserve not only cross-domain consistency and single-domain peculiarity but also multi-typed relations.
- We propose a novel graph-structured CDR framework to jointly model the inter- and intra-domain relations by the relation-aware graph convolutional networks. A gating fusion mechanism is designed to combine domain-shared and domain-specific features for dual-target recommendation.
- We perform extensive experiments on real-world datasets to show the effectiveness of our approaches. The experimental results demonstrate that our approaches significantly outperform the state-of-the-art approaches through detailed analysis.

2 Related Work

2.1 Graph-Based Recommendation

The core idea of graph neural networks (GNNs) is to iteratively aggregate neighbors' features to update the target node feature via propagation mechanism. The mainstream paradigm, such as GCN [14] and GraphSage [9], transforms interactions into a user-item graph to explore the multi-hop connections. GC-MC [1] designs one graph convolutional layer to construct user's and item's embeddings. NGCF [23] explicitly encodes collaborative signals in graph convolutional network (GCN). DGCF [24] disentangles user intents to generate disentangled representations. Furthermore, He et al. [10] propose LightGCN to systematically study several essential components in GCN. However, only considering singular user-item relation makes GNNs impossible to describe users' various preferences.

GNNs can also be used to model heterogeneous graphs constructed from other data. Recent efforts have exhibited incredible performance on many recommendation tasks. GHCF [2] and KHGT [26] design multi-behavior graphs to model multi-typed interactive patterns between users and items. To address the cold-start problem [28], Liu et al. [18] integrate social relation and semantic relation into the user-item graph. Xu et al. [27] utilize interaction sequences to construct a user-subsequence graph and an item-item graph to model user and item similarity on behaviors. Wang et al. [22] incorporate knowledge graphs into sequential recommendation model to enhance user representations.

2.2 Cross-Domain Recommendation

A popular path of CDR systems concentrates on learning a mapping function to transfer knowledge across domains in the same space. Man et al. [20] first introduce the embedding and mapping framework (EMCDR) to achieve cross-domain recommendation step by step. Based on the EMCDR model, CDLFM [25] incorporates users rating behaviors to generate precise latent factors. RC-DFM [7] extend stacked denoising autoencoders to deeply fuse textual contents and ratings. DCDCSR [32] and SSCDR [13] optimize the CDR model to obtain an accurate mapping function. However, the two-stage embedding-and-mapping strategy prevents CDR systems from accomplishing global optimization.

Recently, some methods aim to distill domain-shared features as a bridge between domains and apply symmetrical frameworks to accomplish dual-target CDR. CoNet [12] designs collaborative cross networks to select valuable features for dual knowledge transfer across domains. ACDN [16] integrates users aesthetic features into the CoNet model to transfer domain independent aesthetic preferences. DDTCDR [15] design a latent orthogonal mapping function to capture domain-shared preferences. PPGN [29] and BiTGCF [17] treat overlapped users as shared nodes to construct user-item graphs from ratings, then use GCN to learn cross-domain high-order representations. DA-GCN [8] and π-Net [19] utilize the recurrent neural network and GCN to accomplish the cross-domain sequential recommendation. Differing from existing dual-target CDR works, our method can explicitly transfer knowledge across domains and distill domain-shared features and domain-specific features in a direct way.

3 Preliminaries

Given two specific domains A and B, where the set of users U (of size $M = |U|$) are fully overlapped, the set of items in domains A and B are defined as I (of size $N^A = |I|$) and J (of size $N^B = |J|$), respectively. $u \in U$ indexes a user, $i \in I$ (or $j \in J$) indexes an item from A (or B). The user-item interaction matrices are defined as $\boldsymbol{R}_A \in \mathbb{R}^{M \times N^A}$ in domain A and $\boldsymbol{R}_B \in \mathbb{R}^{M \times N^B}$ in domain B.

Definition 1: Single-domain Heterogeneous Graph. The single-domain graphs $\boldsymbol{G}^A = (\mathcal{V}^U \cup \mathcal{V}^I, \mathcal{E}^A, \mathcal{R}^A)$ and $\boldsymbol{G}^B = (\mathcal{V}^U \cup \mathcal{V}^J, \mathcal{E}^B, \mathcal{R}^B)$ retain the intra-domain relations and user-item relation in A and B, respectively.

Definition 2: Cross-domain Heterogeneous Graph. The cross-domain graph $\boldsymbol{G}^C = (\mathcal{V}^U \cup \mathcal{V}^I \cup \mathcal{V}^J, \mathcal{E}^C, \mathcal{R}^C)$ retains the inter-domain relations and user-item relations within each domain, where $\mathcal{V}^U$, $\mathcal{V}^I$, and $\mathcal{V}^J$ are sets of nodes indicating users, items in A and B, respectively.

Definition 3: Relations. Each edge $e \in \{\mathcal{E}^C, \mathcal{E}^A, \mathcal{E}^B\}$ is associated with the function $\psi(e) : \mathcal{E} \to \mathcal{R}$. $\mathcal{R}$ defines the relations indicating edge types. Formally, we define the type-specific relation $\mathcal{R}_r \in \mathcal{R}$, $r = 1, 2, ..., 8$. $\mathcal{R}_1 =$ {inter-domain user-user relation}, $\mathcal{R}_2 =$ {inter-domain item-item relation}, $\mathcal{R}_3 =$ {intra-domain user-user relation in A}, $\mathcal{R}_4 =$ {intra-domain user-user relation in B}, $\mathcal{R}_5 =$ {intra-domain item-item relation in A}, $\mathcal{R}_6 =$ {intra-domain item-item relation in B}, $\mathcal{R}_7 =$ {user-item relation in A}, $\mathcal{R}_8 =$ {user-item relation in B}. Particularly, $\mathcal{R}^C = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_7, \mathcal{R}_8\}$, $\mathcal{R}^A = \{\mathcal{R}_3, \mathcal{R}_5, \mathcal{R}_7\}$, and $\mathcal{R}^B = \{\mathcal{R}_4, \mathcal{R}_6, \mathcal{R}_8\}$.

Problem Formulation. Our task is to simultaneously predict the probability $\hat{y}^A_{ui}$ and $\hat{y}^B_{uj}$ of unseen interaction between user u, item i in domain A, and item j in domain B, to enhance the accuracy for dual-target recommendation.

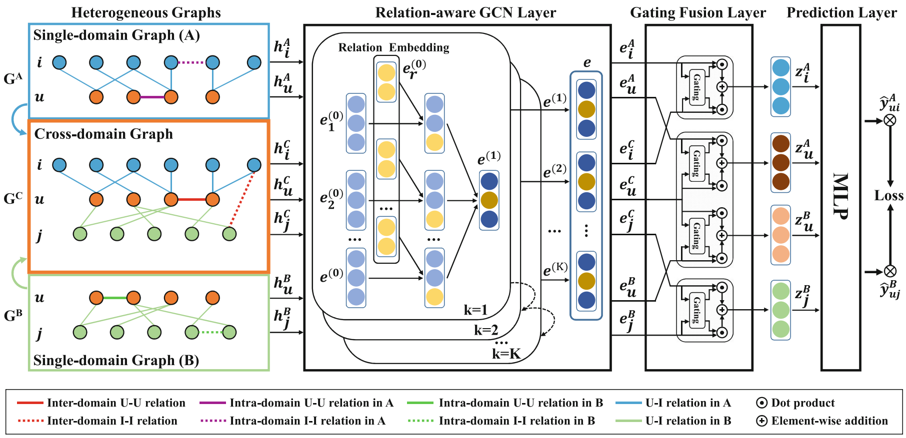


Fig. 2. An illustration of I^2RCDR structure. The U-I relation, U-U relation, I-I relation are short for user-item relation, user-user relation, and item-item relation, respectively.

4 Proposed Framework

Our proposed **I**nter- and **I**ntra-domain **R**elation-aware heterogeneous graph convolutional networks for **C**ross-**D**omain **R**ecommendation (I^2RCDR) is an end-to-end learning framework. The main structure of our proposed framework is illustrated in Fig. 2.

4.1 Graph Construction and Embedding

Inter-domain Relation and Cross-Domain Graph. We first establish the inter-domain user-user relation. For user u, we concatenate all the user's reviews to generate a document d_u. We introduce SentenceBERT [21] to convert the documents d_u into a fixed-size text vector $\boldsymbol{D}_u$, which is formulated as:

$$\boldsymbol{D}_u = SentenceBERT(d_u). \tag{1}$$

To capture the user-user relation between two users across different domains, we compute the cosine similarity to generate the inter-domain semantic link. Particularly, for user u in domain A and u' in domain B, the existing probability score $Pr(u, u')$ of the link between u and u' is as follows:

$$Pr(u, u') = \varphi\left(\frac{\boldsymbol{D}_u \boldsymbol{D}_{u'}}{\|\boldsymbol{D}_u\| \, \|\boldsymbol{D}_{u'}\|}\right), \tag{2}$$

where $Pr(u, u')$ indicates the weight of an edge between u and u'. $\varphi(x) = max(0, x)$ is the ReLU function that normalizes the cosine similarities.

We calculate all the similarities between users from different domains and obtain the inter-domain user-user matrix $\boldsymbol{C}_U \in \mathbb{R}^{M \times M}$. Similarly, we can generate the inter-domain item-item relation matrix $\boldsymbol{C}_I \in \mathbb{R}^{N^A \times N^B}$. With all the matrices formally defined, we then describe the cross-domain graph $\boldsymbol{G}^C$ as shown below:

$$\tilde{\boldsymbol{A}}^C = \begin{bmatrix} \boldsymbol{C}_U & \boldsymbol{R}_A^\top & \boldsymbol{R}_B^\top \\ \boldsymbol{R}_A & \boldsymbol{0} & \boldsymbol{C}_I^\top \\ \boldsymbol{R}_B & \boldsymbol{C}_I & \boldsymbol{0} \end{bmatrix}, \tag{3}$$

where $\tilde{\boldsymbol{A}}^C \in \mathbb{R}^{(M+N^A+N^B)\times(M+N^A+N^B)}$ is the adjacency matrix. $\boldsymbol{R}_A$ and $\boldsymbol{R}_B$ are user-item interaction matrices in domains A and B, respectively. $\boldsymbol{R}_A^\top$, $\boldsymbol{R}_B^\top$, and $\boldsymbol{C}_I^\top$ are the transposed matrices.

Intra-domain Relation and Single-Domain Graph. We explore the user-user (or item-item) relation from domain A to establish the intra-domain user-user relation matrix $\boldsymbol{A}_U$ (or item-item relation matrix $\boldsymbol{A}_I$), which computes the cosine similarities between users (or items) in domain A and gains the probability scores as shown in Eq. (2). We describe the single-domain graph $\boldsymbol{G}^A$ as shown below:

$$\tilde{\boldsymbol{A}}^A = \begin{bmatrix} \boldsymbol{A}_U & \boldsymbol{R}_A^\top \\ \boldsymbol{R}_A & \boldsymbol{A}_I \end{bmatrix}, \tag{4}$$

where $\tilde{\boldsymbol{A}}^A \in \mathbb{R}^{(M+N^A)\times(M+N^A)}$ is the adjacency matrix. Similarly, we obtain the adjacency matrix $\tilde{\boldsymbol{A}}^B$ of graph $\boldsymbol{G}^B$.

Dense Embedding. For user u, item i in domain A, and item j in domain B, we define them using one-hot encodings, namely $\boldsymbol{x}_u \in \mathbb{R}^M$, $\boldsymbol{x}_i \in \mathbb{R}^{N^A}$, and $\boldsymbol{x}_j \in \mathbb{R}^{N^B}$. For relation $\mathcal{R}_r$ which indexes edge type, we also define $\boldsymbol{x}_r \in \mathbb{R}^8$. Then, we map the one-hot encodings into dense embeddings as follows:

$$\boldsymbol{h}_u = \boldsymbol{P}_u\boldsymbol{x}_u, \boldsymbol{h}_i = \boldsymbol{P}_i\boldsymbol{x}_i, \boldsymbol{h}_j = \boldsymbol{P}_j\boldsymbol{x}_j, \boldsymbol{h}_r = \boldsymbol{P}_r\boldsymbol{x}_r, \tag{5}$$

where $\boldsymbol{P}_u = \{\boldsymbol{P}_u^C, \boldsymbol{P}_u^A, \boldsymbol{P}_u^B\} \in \mathbb{R}^{(M+N^A+N^B)\times d}$, $\boldsymbol{P}_i = \{\boldsymbol{P}_i^C, \boldsymbol{P}_i^A\} \in \mathbb{R}^{(M+N^A)\times d}$, $\boldsymbol{P}_j = \{\boldsymbol{P}_j^C, \boldsymbol{P}_j^B\} \in \mathbb{R}^{(M+N^B)\times d}$ and $\boldsymbol{P}_r \in \mathbb{R}^{8\times d}$ are transformation matrices. d denotes the embedding size. $\boldsymbol{h}_u = \{\boldsymbol{h}_u^C, \boldsymbol{h}_u^A, \boldsymbol{h}_u^B\}$ indicate embeddings of u in $\boldsymbol{G^C}$, $\boldsymbol{G^A}$, and $\boldsymbol{G^B}$, respectively. $\boldsymbol{h}_i = \{\boldsymbol{h}_i^C, \boldsymbol{h}_i^A\}$ indicate embeddings of i in $\boldsymbol{G^C}$ and $\boldsymbol{G^A}$. $\boldsymbol{h}_j = \{\boldsymbol{h}_j^C, \boldsymbol{h}_j^B\}$ indicate embeddings of j in $\boldsymbol{G^C}$ and $\boldsymbol{G^B}$.

4.2 Relation-Aware GCN Layer

Next, we introduce GCN to capture relation-aware multi-hop heterogeneous connections between users and items in different graphs. Considering different types of relations between nodes, we compose a neighboring node with respect to its relation to model a relation-aware target node. Specifically, we incorporate the relation embeddings into the propagation process via element-wise addition. We take user u as an example and formulate relation-aware GCN as follows:

$$\boldsymbol{e}_u^{(k+1)} = \sigma\left(\sum_{(v_r)\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_{v_r}|}} \boldsymbol{W}^{(k)} \left(\boldsymbol{e}_{v_r}^{(k)} \oplus \boldsymbol{e}_r^{(k)}\right)\right), \tag{6}$$

where $\boldsymbol{e}_{v_r}^{(k)}$ and $\boldsymbol{e}_r^{(k)}$ respectively denote the embeddings of node v_r and relation r after k layers propagation. σ is the nonlinear activate function and $\boldsymbol{W}^{(k)}$ is the weight matrix. $\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_{v_r}|}}$ is a symmetric normalization constant. $\mathcal{N}_u$ and $\mathcal{N}_{v_r}$ denote the neighbors of u and v_r. v_r are the neighbors of user u under relation type $\mathcal{R}_r$. For example, in graph $\boldsymbol{G}^C$, the neighbors $v_r \in \{v_1, v_7, v_8\}$ correspond to three type-specific relations $\{\mathcal{R}_1, \mathcal{R}_7, \mathcal{R}_8\}$. $\oplus$ denotes the element-wise addition. Note that other composition ways like element-wise product can also be used, we leave it for future research. $\boldsymbol{e}_r^{(k)}$ defines the relation embeddings, which is updated as follows:

$$\boldsymbol{e}_r^{(k+1)} = \boldsymbol{W}_r^{(k)} \boldsymbol{e}_r^{(k)}, \tag{7}$$

where $\boldsymbol{W}_r^{(k)}$ is a weight matrix which maps relations to the same space as nodes. Note that $\boldsymbol{h}_u$, $\boldsymbol{h}_i$, $\boldsymbol{h}_j$ and $\boldsymbol{h}_r$ are defined as initial embeddings $\boldsymbol{e}_u^{(0)}$, $\boldsymbol{e}_i^{(0)}$, $\boldsymbol{e}_j^{(0)}$, and $\boldsymbol{e}_r^{(0)}$ for node u, i, j, and relation $\mathcal{R}_r$, respectively.

To offer a easy-to-understand perspective of propagation mechanism, we use matrix form to describe this propagation process (equivalent to Eq. (6)):

$$\boldsymbol{E}_u^{(k+1)} = \sigma\left(\hat{A}(\boldsymbol{E}_{v_r}^{(k)} \oplus \boldsymbol{E}_r^{(k)}) \boldsymbol{W}^{(k)}\right), \tag{8}$$

where $\boldsymbol{E}_{v_r}^{(k)}$ and $\boldsymbol{E}_r^{(k)}$ are the embeddings of v_r and corresponding relation $\mathcal{R}_r$ obtained after k steps of propagation. $\hat{A} = \boldsymbol{D}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \boldsymbol{D}^{-\frac{1}{2}}$ denotes the symmetrically normalized matrix. $\tilde{\boldsymbol{A}} \in \{\tilde{\boldsymbol{A}}^C, \tilde{\boldsymbol{A}}^A, \tilde{\boldsymbol{A}}^B\}$ and $\boldsymbol{D}$ is the diagonal degree matrix of $\tilde{\boldsymbol{A}}$.

After aggregating and propagating with K steps, we generate and stack multiple embeddings of user u, namely $\left\{\boldsymbol{e}_u^{(0)}, \boldsymbol{e}_u^{(1)}, \cdots, \boldsymbol{e}_u^{(K)}\right\}$. The combination of embeddings refined from different order neighbors can better indicate the features of user u. As such, we concatenate different embeddings with the following formula and get the final embeddings of user u.

$$\boldsymbol{e}_u = \boldsymbol{e}_u^{(0)} \| \cdots \| \boldsymbol{e}_u^{(k)} \| \cdots \| \boldsymbol{e}_u^{(K)}, \tag{9}$$

where $\|$ denotes concatenation operation. $\boldsymbol{e}_u^{(k)}$ is the embeddings of user u with k steps, $k = 0, 1, 2, ..., K$. $\boldsymbol{e}_u \in \mathbb{R}^{d'}$ is the final embeddings of user u, d' denotes the embedding size after K steps of propagation and concatenation.

In this layer, we can utilize the relation-aware GCN to generate not only domain-shared features $\boldsymbol{e}_u^C$ of user u in $\boldsymbol{G}^C$ but also domain-specific features $\boldsymbol{e}_u^A$ (or $\boldsymbol{e}_u^B$) of user u in $\boldsymbol{G}^A$ (or $\boldsymbol{G}^B$). Similarly, we obtain domain-shared features $\boldsymbol{e}_i^C$ of item i in $\boldsymbol{G}^C$, domain-shared features $\boldsymbol{e}_j^C$ of item j in $\boldsymbol{G}^C$, domain-specific features $\boldsymbol{e}_i^A$ of item i in $\boldsymbol{G}^A$, and domain-specific features $\boldsymbol{e}_j^B$ of item j in $\boldsymbol{G}^B$.

4.3 Gating Fusion Layer

The relation-aware GCN can simultaneously model the inter- and intra-domain relations to distill domain-shared and domain-specific features. Therefore, the combination of two types of features needs to balance the cross-domain consistency and single-domain peculiarity. Motivated by the milestone work Gated Recurrent Unit (GRU) [4], we introduce the gating mechanism to differentiate the importance of domain-shared and domain-specific features for dual-target recommendation. To combine user's features $\boldsymbol{e}_u^C$, $\boldsymbol{e}_u^A$, and $\boldsymbol{e}_u^B$, we use two neural gating units that share partial parameters to generate two combination features of domains A and B. The process is computed as:

$$G_u^A = \text{sigmoid}\left(\boldsymbol{V}_u^C \boldsymbol{e}_u^C + \boldsymbol{V}_u^A \boldsymbol{e}_u^A\right), \tag{10}$$

$$G_u^B = \text{sigmoid}\left(\boldsymbol{V}_u^C \boldsymbol{e}_u^C + \boldsymbol{V}_u^B \boldsymbol{e}_u^B\right), \tag{11}$$

$$\boldsymbol{z}_u^A = G_u^A \odot \boldsymbol{e}_u^C + (1 - G_u^A) \odot \boldsymbol{e}_u^A, \tag{12}$$

$$\boldsymbol{z}_u^B = G_u^B \odot \boldsymbol{e}_u^C + (1 - G_u^B) \odot \boldsymbol{e}_u^B, \tag{13}$$

where $\boldsymbol{V}_u^C \in \mathbb{R}^{d' \times d'}$ is a shared weight matrix for the two gating units, $\boldsymbol{V}_u^A$ and $\boldsymbol{V}_u^B \in \mathbb{R}^{d' \times d'}$ are weight matrices. $\boldsymbol{z}_u^A$ and $\boldsymbol{z}_u^B$ indicate the combination features of user u in A and B, respectively. The shared parameters $\boldsymbol{V}_u^C$ make two combination features assigned to the same weight for bidirectional knowledge transfer, ensuring the consistency of domain-shared part.

For the features $\boldsymbol{e}_i^C$ and $\boldsymbol{e}_i^A$ of item i in domain A, we apply a standard gating unit to combine them adaptively with the following formula:

$$G_i^A = \text{sigmoid}\left(\boldsymbol{V}_i^C \boldsymbol{e}_i^C + \boldsymbol{V}_i^A \boldsymbol{e}_i^A\right), \tag{14}$$

$$\boldsymbol{z}_i^A = G_i^A \odot \boldsymbol{e}_i^C + (1 - G_i^A) \odot \boldsymbol{e}_i^A, \tag{15}$$

where $\boldsymbol{V}_i^C$ and $\boldsymbol{V}_i^A \in \mathbb{R}^{d' \times d'}$ are weight matrices. Similarly, we can gain the combined feature $\boldsymbol{z}_j^B$ of item j in domain B.

4.4 Prediction Layer

After the combination of features, we generate user features ($\boldsymbol{z}_u^A$ and $\boldsymbol{z}_u^B$) and item features ($\boldsymbol{z}_i^A$ and $\boldsymbol{z}_j^B$). To endow the CDR systems with non-linearity, we apply the multi-layer perception (MLP) to model the user-item interactions. The formula in domain A is as follows:

$$\phi^1 = a^1(\boldsymbol{S}^1 \begin{bmatrix} \boldsymbol{z}_u^A \\ \boldsymbol{z}_i^A \end{bmatrix} + \boldsymbol{b}^1), \tag{16}$$

$$\ldots\ldots \tag{17}$$

$$\phi^L = a^L(\boldsymbol{S}^L \phi^{L-1} + \boldsymbol{b}^L), \tag{18}$$

$$\hat{y}_{ui}^A = f(\phi^L), \tag{19}$$

where $\boldsymbol{S}^l$ and $\boldsymbol{b}^l$ denote the trainable matrix and bias term for the l-th layer, respectively. a^l denotes the activation function such as sigmoid, ReLU, and hyperbolic tangent (tanh). ϕ^l denotes the output result for the l-th layer. $f(\cdot)$ is the prediction function, which maps ϕ^L to the probability $\hat{y}_{ui}^A$. Analogously, we generate the probability score $\hat{y}_{uj}^B$ in domain B.

4.5 Model Training

For CDR systems, an appropriate loss function can make the model achieve global optimization and speed up the model convergence. Considering the nature of implicit feedback, we select cross-entropy as the loss function which is defined as follows:

$$\mathcal{L}\left(\hat{y}_{uv}, y_{uv}\right) = -\sum_{(u,v)\in\mathcal{P}^{+}\cup\mathcal{P}^{-}} y_{uv}\log\hat{y}_{uv} + \left(1 - y_{uv}\right)\log\left(1 - \hat{y}_{uv}\right), \tag{20}$$

where y_{uv} defines an observed interaction and $\hat{y}_{uv}$ defines its corresponding predicted interaction. $\mathcal{P}^{+}$ is the set of observed interactions, $\mathcal{P}^{-}$ is a certain number of negative instances that can be randomly sampled from unobserved interaction to prevent over-fitting.

We aim to simultaneously enhance the performance of recommendation in both domains. Hence, the joint loss function to be minimized for domain A ($\mathcal{L}_A$) and domain B ($\mathcal{L}_B$) is defined as:

$$\mathcal{L}_{joint} = \alpha\mathcal{L}_A + \beta\mathcal{L}_B + \mathcal{L}_{reg} = \alpha\mathcal{L}(\hat{y}^{A}_{ui}, y^{A}_{ui}) + \beta\mathcal{L}(\hat{y}^{B}_{uj}, y^{B}_{uj}) + \gamma\|\Theta\|_2^2, \tag{21}$$

where $\mathcal{L}(\hat{y}^{A}_{ui}, y^{A}_{ui})$ and $\mathcal{L}(\hat{y}^{B}_{uj}, y^{B}_{uj})$ define the loss function in domains A and B, respectively. Considering that the sparseness of interactions in the two domains is inconsistent, we leverage α and β to control sample balance. Here, we set $\alpha = \beta = 1$, considering two recommendation tasks for domains A and B are of equal importance. $\mathcal{L}_{reg}$ is a regularization term, in which γ is a hyper-parameter that controls the importance of $L2$ regularization and Θ are network parameters.

5 Experiments

This section answers the following questions:
RQ1: How does I^2RCDR perform compared with baselines?
RQ2: How do different designed modules (i.e., relation-aware GCN, gating fusion mechanism, and MLP) contribute to the model performance?
RQ3: Do inter-domain relations provide valuable information?
RQ4: How does I^2RCDR perform with different parameter settings?

5.1 Experimental Settings

Dataset. We examine the performance of our CDR framework on the real-world and well-known Amazon dataset[1], which includes abundant rating and review data and is widely used in CDR systems. We choose three pairs of datasets to organize our experiments. We first transform the ratings into implicit data, where each interaction is marked as 0 or 1, indicating whether the user has rated the item. We then select overlapped users from each pair of datasets and filter out non-overlapped users. Table 1 summarizes the detailed statistics of datasets.

[1] http://jmcauley.ucsd.edu/data/amazon/.

Table 1. Statistics of datasets.

Dataset	# Users	# Items	# Interactions	Density
Toys and Games (Toy)	1,380	6,773	19,831	0.212%
Video Games (Video)	1,380	6,667	21,359	0.232%
Sports and Outdoors (Sport)	3,908	13,057	43,996	0.086%
Clothing Shoes and Jewelry (Cloth)	3,908	13,044	35,115	0.069%
Home and Kitchen (Home)	14,059	25,995	179,543	0.049%
Health and Personal Care (Health)	14,059	17,663	174,998	0.071%

Evaluation Metric. We employ the *leave-one-out* strategy to conduct the evaluation. Since we focus on the top-N recommendation tasks, we apply two widely adopted metrics, namely Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [13], to efficiently estimate the performance of our method and baselines. For each user, we randomly sample 99 negative items that have not been rated with the user and combine them with the positive instance the user has been rated as the list waiting to be sorted in the ranking procedure. We repeat this process 5 times and show the average ranking results.

Comparison Methods. We compare our proposed method with state-of-the-art approaches and categorize the baselines into three groups: Single-Domain Recommendation (SDR), MLP-based CDR, and GNN-based CDR.

- **SDR.** GC-MC [1] applies GNNs to recommendation tasks and converts rating user-item interaction into a bipartite graph. NGCF [23] explicitly injects collaborative signals into the user-item graph. LightGCN [10] researches three different components in GCN and demonstrates neighbor aggregation is the most essential factor. HGNR [18] uses ratings, reviews, and social network data to construct a heterogeneous graph.
- **MLP-based CDR.** MLP-based CDR mainly introduces MLP to learn hidden features. CoNet [12] designs cross-connection networks to achieve dual knowledge transfer based on the cross-stitch network model. DTCDR [31] proposes a dual-target CDR framework to integrate the domain-shared features of overlapped users from two domains.
- **GNN-based CDR.** This group applies GNN to learn high-order features from graphs. PPGN [29] fuses user-item graphs from two domains into a holistic graph. BiTGCF [17] establishes a user-item graph for each domain and uses the overlapped user as the bridge to fuse users domain-shared and domain-specific features. GA-DTCDR [33] designs two independent heterogeneous graphs which are constructed from ratings and reviews.

Parameter Settings. We utilize Tensorflow to implement our framework and all the baselines. In our model learning stage, we choose *Adam* as the optimizer

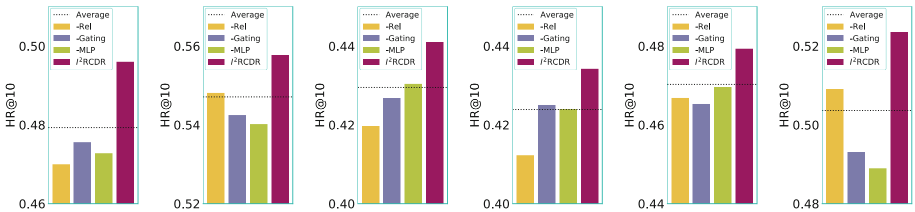
Table 2. Recommendation performance of compared methods in terms of HR and NDCG. The best performance is in boldface and the best baseline is underlined.

Dataset	Toy & Video				Home & Hearth				Sport & Cloth			
	Toy		Video		Home		Hearth		Sport		Cloth	
Method	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
GC-MC	0.276	0.135	0.317	0.154	0.284	0.138	0.275	0.136	0.371	0.194	0.385	0.201
NGCF	0.295	0.156	0.370	0.190	0.326	0.169	0.312	0.162	0.393	0.212	0.412	0.221
LightGCN	0.313	0.159	0.397	0.205	0.300	0.146	0.289	0.135	0.372	0.189	0.386	0.191
HGNR	0.323	0.170	0.405	0.224	0.356	0.202	0.311	0.164	0.404	0.234	0.410	0.233
CoNet	0.375	0.229	0.476	0.275	0.366	0.213	0.318	0.193	0.417	0.224	0.420	0.196
DTCDR	0.455	0.254	0.518	0.294	0.394	0.210	0.382	0.195	0.444	0.253	0.471	0.268
PPGN	0.457	0.265	0.498	0.285	0.396	0.174	0.386	0.197	0.446	0.234	0.460	0.228
BiTGCF	0.443	0.231	0.524	0.297	0.397	0.229	0.346	0.181	0.456	0.264	0.474	0.274
GA-DTCDR	0.464	0.254	0.521	0.302	0.425	0.226	0.398	0.212	0.459	0.266	0.481	0.286
I^2**RCDR**	**0.496**	**0.285**	**0.558**	**0.342**	**0.441**	**0.252**	**0.434**	**0.246**	**0.479**	**0.281**	**0.524**	**0.297**

for all models to update model parameters and set the initial learning rate as 0.001. We sample four negative instances for each positive instance to generate the training dataset. The batch size is set to 512. we set $d = 32$ for the embedding size of all the methods. Furthermore, we use dropout techniques to further prevent over-fitting and fix the dropout rate as 0.1. For our proposed method and GNN-based baselines, we set $k = 4$. For HGNR, we use reviews to construct user-user graph instead of the social network graph. For DTCDR and GA-DTCDR, we only consider the set of overlapped uses and model the review information to ensure fairness.

5.2 RQ1: Performance Comparison

Table 2 reports the summarized results of our experiments on three pairs of datasets in terms of HR@10 (H@10) and NDCG@10 (N@10). It can be seen that I^2RCDR consistently achieves the best performance compared with all the baselines, which reveals the superiority of modeling inter-domain and intra-domain relations collectively by relation-aware GCN. Compared with SDR approaches (GC-MC, NGCF, LightGCN, and HGNR), CDR models usually obtain better performance, benefit from fusing more useful knowledge from both two domains during the transfer learning phase. In CDR methods, GNN-based models (PPGN, BiTGCF, and GA-DTCDR) outperform MLP-based models (CoNet and DTCDR) by 6.28% H@10 and 4.72% N@10 on average, respectively. This observation justifies that GNNs can achieve better recommendation performance by modeling the high-order representations. Furthermore, DTCDR and GA-DTCDR relying on integrating review information into CDR systems, are indeed better than other CDR models which only use rating information, but are still weaker than our method which fuses reviews and ratings to learn domain-shared and domain-specific features explicitly.



(a) Toy (b) Video (c) Sport (d) Cloth (e) Home (f) Health

Fig. 3. Performance of I^2RCDR compared with different variants in terms of HR@10

5.3 RQ2: Ablation Study

We attempt to validate whether I^2RCDR benefits from the influence of different modules. Therefore, we compare I^2RCDR with the following variant versions:

- **-Rel.** This method performs a typical GCN model [14] instead of our relation-aware GCN.
- **-Gating.** This variant employs an element-wise attention mechanism to replace our gating fusion mechanism.
- **-MLP.** This method removes MLP in the prediction layer and defines an inner product as the interaction function.

As presented in Fig. 3, we observe that our I^2RCDR is better than all the variants in terms of HR@10. We overlook the performance of NDCG which follows the similar trend due to the space limitation.

5.4 RQ3: Effect of Inter-domain Relations

To verify the effectiveness of inter-domain user-user and item-item relations, we compare the performance of our method leveraging one or two kinds of relations in terms of HR@10 and NDCG@10. Figure 4 illustrates the comparison results concerning I^2RCDR-UI (without considering inter-domain relations), I^2RCDR-I (only considering user-user relation), I^2RCDR-U (only considering item-item relation), I^2RCDR (simultaneously considering two kinds of relations). I^2RCDR-I and I^2RCDR-U both have better performance by adding inter-domain user-user or item-item relation than I^2RCDR. This proves the effectiveness of integrating inter-domain relations in CDR tasks. In addition, we ignore the results about the effect of intra-domain relations due to space limitations, which exhibit a similar pattern to inter-domain relations.

5.5 RQ4: Parameter Analysis

We vary the layer numbers which are in the range of {2, 3, 4, 5, 6}, to verify whether I^2RCDR benefits from multiple propagation layers. Figure 5 shows the results about different layers on three pairs of datasets in terms of HR@10. When

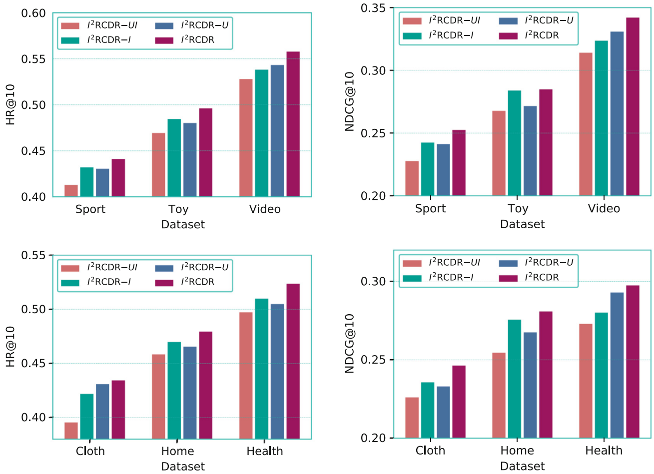


Fig. 4. Performance of I^2RCDR with different inter-domain relations

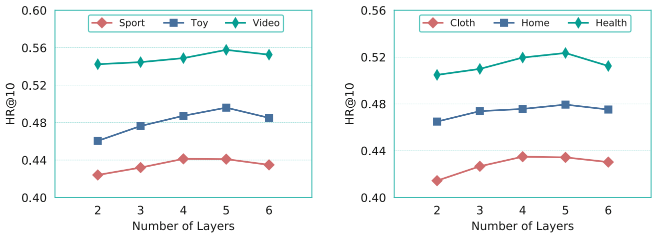


Fig. 5. Effect of propagation layer numbers

the number of layers is 4 or 5, we obtain the finest results. The performance significantly improves as the number of layers increases. We think this is because more potential relations have been mined with the increasing of model depth. However, the results also show that longer layers (e.g., 6-hop) maybe make a lot of noise, which affects the recommendation accuracy.

6 Conclusion and Future Work

In this paper, we propose I^2RCDR, an end-to-end graph-structured framework that naturally incorporates inter-domain relations into CDR systems. I^2RCDR designs the relation-aware GCNs to encode heterogeneous graphs and jointly model inter- and intra-domain relations. To balance cross-domain consistency and single-domain peculiarity, we design a gating fusion mechanism to fuse domain-shared and domain-specific features for dual-target recommendation. Extensive experiments are carried out on three pairs of datasets and the results demonstrate the effectiveness of the proposed framework. Currently, we only deal with the holistic user-user and item-item relations produced from reviews. In future work, we will consider disentangling semantic relations to extract multifaceted features and model users' fine-grained preferences.

Acknowledgments. This research is supported in part by the 2030 National Key AI Program of China 2018AAA0100503 (2018AAA0100500), National Science Foundation of China (No. 62072304, No. 61772341, No. 61832013), Shanghai Municipal Science and Technology Commission (No. 19510760500, No. 19511101500, No. 19511120300), the Oceanic Interdisciplinary Program of Shanghai Jiao Tong University (No. SL2020MS032), Scientific Research Fund of Second Institute of Oceanography, the open fund of State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, MNR, and Zhejiang Aoxin Co. Ltd.

References

1. van den Berg, R., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263 (2017)
2. Chen, C., et al.: Graph heterogeneous multi-relational recommendation. In: AAAI, vol. 35, pp. 3958–3966 (2021)
3. Chen, C., et al.: An efficient adaptive transfer neural network for social-aware recommendation. In: SIGIR, pp. 225–234 (2019)
4. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
5. Cui, Q., Wei, T., Zhang, Y., Zhang, Q.: HeroGRAPH: a heterogeneous graph framework for multi-target cross-domain recommendation. In: ORSUM@ RecSys (2020)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Fu, W., Peng, Z., Wang, S., Xu, Y., Li, J.: Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In: AAAI, vol. 33, pp. 94–101 (2019)
8. Guo, L., Tang, L., Chen, T., Zhu, L., Nguyen, Q.V.H., Yin, H.: DA-GCN: a domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. arXiv preprint arXiv:2105.03300 (2021)
9. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS, pp. 1025–1035 (2017)
10. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. arXiv preprint arXiv:2002.02126 (2020)
11. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW, pp. 173–182 (2017)
12. Hu, G., Zhang, Y., Yang, Q.: CoNET: collaborative cross networks for cross-domain recommendation. In: CIKM, pp. 667–676 (2018)
13. Kang, S., Hwang, J., Lee, D., Yu, H.: Semi-supervised learning for cross-domain recommendation to cold-start users. In: CIKM, pp. 1563–1572 (2019)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

15. Li, P., Tuzhilin, A.: DDTCDR: deep dual transfer cross domain recommendation. In: WSDM, pp. 331–339 (2020)
16. Liu, J., et al.: Exploiting aesthetic preference in deep cross networks for cross-domain recommendation. In: WWW, pp. 2768–2774 (2020)
17. Liu, M., Li, J., Li, G., Pan, P.: Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In: CIKM, pp. 885–894 (2020)
18. Liu, S., Ounis, I., Macdonald, C., Meng, Z.: A heterogeneous graph neural model for cold-start recommendation. In: SIGIR, pp. 2029–2032 (2020)
19. Ma, M., Ren, P., Lin, Y., Chen, Z., Ma, J., de Rijke, M.: π-Net: a parallel information-sharing network for shared-account cross-domain sequential recommendations. In: SIGIR, pp. 685–694 (2019)
20. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: an embedding and mapping approach. In: IJCAI, vol. 17, pp. 2464–2470 (2017)
21. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)
22. Wang, C., Zhu, Y., Liu, H., Ma, W., Zang, T., Yu, J.: Enhancing user interest modeling with knowledge-enriched itemsets for sequential recommendation. In: CIKM, pp. 1889–1898 (2021)
23. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: SIGIR, pp. 165–174 (2019)
24. Wang, X., Jin, H., Zhang, A., He, X., Xu, T., Chua, T.S.: Disentangled graph collaborative filtering. In: SIGIR, pp. 1001–1010 (2020)
25. Wang, X., Peng, Z., Wang, S., Yu, P.S., Fu, W., Hong, X.: Cross-domain recommendation for cold-start users via neighborhood based feature mapping. In: Pei, J., Manolopoulos, Y., Sadiq, S., Li, J. (eds.) DASFAA 2018. LNCS, vol. 10827, pp. 158–165. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91452-7_11
26. Xia, L., et al.: Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: AAAI, vol. 35, pp. 4486–4493 (2021)
27. Xu, Y., Zhu, Y., Shen, Y., Yu, J.: Learning shared vertex representation in heterogeneous graphs with convolutional networks for recommendation. In: IJCAI, pp. 4620–4626 (2019)
28. Zang, T., Zhu, Y., Liu, H., Zhang, R., Yu, J.: A survey on cross-domain recommendation: taxonomies, methods, and future directions. arXiv preprint arXiv:2108.03357 (2021)
29. Zhao, C., Li, C., Fu, C.: Cross-domain recommendation via preference propagation GraphNet. In: CIKM, pp. 2165–2168 (2019)
30. Zhao, C., Li, C., Xiao, R., Deng, H., Sun, A.: CATN: cross-domain recommendation for cold-start users via aspect transfer network. In: SIGIR, pp. 229–238 (2020)
31. Zhu, F., Chen, C., Wang, Y., Liu, G., Zheng, X.: DTCDR: a framework for dual-target cross-domain recommendation. In: CIKM, pp. 1533–1542 (2019)
32. Zhu, F., Wang, Y., Chen, C., Liu, G., Orgun, M., Wu, J.: A deep framework for cross-domain and cross-system recommendations. arXiv preprint arXiv:2009.06215 (2020)
33. Zhu, F., Wang, Y., Chen, C., Liu, G., Zheng, X.: A graphical and attentional framework for dual-target cross-domain recommendation. In: IJCAI, pp. 3001–3008 (2020)