A Graphical and Attentional Framework for Dual-Target Cross-Domain Recommendation

Feng Zhu[1] , **Yan Wang**[1] [*] , **Chaochao Chen**[2] , **Guanfeng Liu**[1] and **Xiaolin Zheng**[3]
[1] Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
[2] Ant Financial Services Group, Hangzhou 310012, China
[3] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
feng.zhu3@hdr.mq.edu.au, yan.wang@mq.edu.au, chaochao.ccc@antfin.com, guanfeng.liu@mq.edu.au, xlzheng@zju.edu.cn

Abstract

The conventional single-target Cross-Domain Recommendation (CDR) only improves the recommendation accuracy on a target domain with the help of a source domain (with relatively richer information). In contrast, the novel dual-target CDR has been proposed to improve the recommendation accuracies on both domains simultaneously. However, dual-target CDR faces two new challenges: (1) how to generate more representative user and item embeddings, and (2) how to effectively optimize the user/item embeddings on each domain. To address these challenges, in this paper, we propose a graphical and attentional framework, called GA-DTCDR. In GA-DTCDR, we first construct two separate heterogeneous graphs based on the rating and content information from two domains to generate more representative user and item embeddings. Then, we propose an element-wise attention mechanism to effectively combine the embeddings of common users learned from both domains. Both steps significantly enhance the quality of user and item embeddings and thus improve the recommendation accuracy on each domain. Extensive experiments conducted on four real-world datasets demonstrate that GA-DTCDR significantly outperforms the state-of-the-art approaches.

1 Introduction

Targeting data sparsity problem, Cross-Domain Recommendation (CDR) [Berkovsky *et al.*, 2007] aims to leverage the richer information from a richer domain to only help improve the recommendation accuracy on a sparser domain, resulting in *single-target CDR*. In contrast, the novel *dual-target CDR* has been recently proposed to improve the recommendation accuracies on both richer and sparser domains simultaneously by making good use of the information or knowledge from both domains [Zhu *et al.*, 2019; Li and Tuzhilin, 2019].

Intuitively, based on the existing single-target CDR approaches, it seems to be a solution for dual-target CDR by simply changing their transfer direction from "Richer→Sparser" to "Sparser→Richer". However, as referred to as *Negative Transfer* [Pan and Yang, 2009], this idea does not work, because, in principle, the knowledge learned from the sparser domain is less accurate than that learned from the richer domain, and thus the recommendation accuracy on the richer domain is more likely to decline by simply and directly changing the transfer direction. Therefore, dual-target CDR demands novel and effective solutions.

To achieve good-performing dual-target CDR, there are two significant challenges in the literature as follows.

CH1. *how to leverage the data richness and diversity to generate more representative single-domain user and item embeddings for improving recommendation accuracy on each of both domains?* Both traditional Collaborative Filtering (CF) models, e.g., BPR [Rendle *et al.*, 2009], and novel neural CF models, e.g., NeuMF [He *et al.*, 2017] and DMF [Xue *et al.*, 2017], are based on the user-item relationship to learn user and item embeddings. However, most of them ignore the user-user and item-item relationships, and thus can hardly enhance the quality of embeddings.

CH2. *how to effectively optimize the user or item embeddings on each target domain for improving recommendation accuracies on both domains?* The state-of-the-art dual-target CDR approaches either adopt fixed combination strategies, e.g., average-pooling, max-pooling, and concatenation [Zhu *et al.*, 2019], or simply adapt the existing single-target transfer learning to dual transfer learning [Li and Tuzhilin, 2019]. However, none of them can effectively combine the embeddings of common users/items, and thus it is hard to achieve an effective embedding optimization on each target domain.

Our Approach and Contributions. For the above two challenges, in this paper, we propose a novel graphical and attentional approach for dual-target CDR. The characteristics and contributions of our work are summarized as follows:

- We propose a **G**raphical and **A**ttentional framework for **D**ual-**T**arget **C**ross-**D**omain **R**ecommendation, called GA-DTCDR, which can leverage the data richness and diversity (e.g., ratings, reviews, and tags) of dual domains, share the knowledge of common users across domains, and make recommendations on both domains;
- To address **CH1**, we construct a heterogeneous graph, considering not only user-item relationships (based on ratings),

[*] Contact Author

but also user-user and item-item relationships (based on content similarities). Then, with this heterogeneous graph, we apply a graph embedding technique, i.e., Node2vec, to generate more representative single-domain user and item embeddings for accurately capturing user and item features;

- To address **CH2**, we propose an element-wise attention mechanism to effectively combine the embeddings of common users learned from both domains. This mechanism trains two separate element-wise attention networks for the two target domains respectively, which can significantly enhance the quality of user embeddings and thus improve the recommendation accuracy on each of two domains simultaneously;
- We conduct extensive experiments on four real-world datasets, which demonstrate that our GA-DTCDR significantly outperforms the best-performing baselines by an average of 8.46% in terms of recommendation accuracy.

2 Related Work

2.1 Single-Target and Dual-Target CDR

Single-Target CDR. According to transfer strategies, the existing single-target CDR approaches can be generally classified in two categories, i.e., content-based transfer and feature-based transfer. *Content-based Transfer* approaches first link the richer and sparser domains by content information, such as attributes [Berkovsky *et al.*, 2007] and text [Tan *et al.*, 2014], then share user/item knowledge across domains. *Feature-based Transfer* approaches tend to apply some classical machine learning techniques, e.g., multi-task learning [Lu *et al.*, 2018], transfer learning [Hu *et al.*, 2019], and neural networks [Man *et al.*, 2017; Zhu *et al.*, 2018; Fu *et al.*, 2019], to directly share or indirectly map user/item embeddings or rating patterns [Yuan *et al.*, 2019] across domains. Also, some studies [Pan and Yang, 2013; Zhang *et al.*, 2016] focus on a derivational problem, i.e., multi-domain recommendation, which is to improve the recommendation accuracy on the target domain by leveraging the auxiliary information from multiple domains. However, all of them are single-target models, which means they cannot improve the recommendation accuracy on the richer domain even if the sparser domain may contain certain types of auxiliary information to support the richer domain.

Dual-Target CDR. The existing dual-target CDR approaches mainly focus on applying fixed combination strategies, e.g., average-pooling, max-pooling, and concatenation [Zhu *et al.*, 2019], or simply improving the existing single-target transfer learning to be a dual transfer learning [Li and Tuzhilin, 2019]. However, none of them can effectively combine the embeddings of common users.

2.2 Graph Embedding

Graph Embedding is to learn a mapping function which maps the nodes in a graph to low-dimensional latent representations [Zhou *et al.*, 2018; Liu *et al.*, 2019]. According to embedding techniques, we review the existing graph embedding approaches into two categories, i.e., dimensionality reduction and neural network. *Dimensionality reduction-based* approaches [Wold *et al.*, 1987] optimize a function that reduces the dimension of a graph matrix and then produce low-dimensional embeddings. *Neural network-based* approaches, e.g., Node2vec [Grover and Leskovec, 2016], treat nodes as words and the generated random walks on graphs as sentences, and then learn node embeddings based on these words and sentences.

Symbol	**Definition**
$c_{ij} \in C$	the comment (e.g., the review and the tags) of user u_i on item v_j
$C \in \mathbb{R}^{m \times n}$	the user comments
$D = \{d_1, d_2, ..., d_{m+n}\}$	the content documents of users and items
$ID = \{id_1, ..., id_n\}$	the item details
$G = (\{\mathcal{U}, \mathcal{V}\}, E)$	the heterogeneous graph, E is the set of user-user, user-item, and item-item relationships
k	the dimension of embedding matrix
m	the number of users
n	the number of items
$\tilde{U}$	the combined embeddings of common users
$r_{ij} \in R$	the rating of user u_i on item v_j
$R \in \mathbb{R}^{m \times n}$	the rating matrix
$\mathcal{U} = \{u_1, ..., u_m\}$	the set of users
U	the graph embedding matrix of users
UC	the document embedding matrix of users
$UP = \{up_1, ..., up_m\}$	the user profiles
$\mathcal{V} = \{v_1, ..., v_n\}$	the set of items
V	the graph embedding matrix of items
VC	the document embedding matrix of items
$y_{ij} \in Y$	the interaction of user u_i on item v_j
$Y \in \mathbb{R}^{m \times n}$	the user-item interaction matrix
$*^a$ and $*^b$	the notations for domains a and b, e.g., m^a represents the number of users on domain a
$\hat{*}$	the predicted notations, e.g., $\hat{y}_{ij}$ represents the predicted interaction of u_i on item v_j

Table 1: Important notations

2.3 Attention Mechanism

Attention is firstly introduced in [Bahdanau *et al.*, 2014], which provides more accurate alignment for each position in a machine translation task. Apart from machine translation, recently, attention mechanism also has been widely used in recommendation [Chen *et al.*, 2017]. The general idea of attention mechanism is to focus on selective parts of the whole information, which can capture the outstanding features of objects. For recommendation, the existing attention approaches [Wang *et al.*, 2019] tend to select more informative parts of explicit or implicit data to improve the representations for users and items.

3 The Proposed Model

In this section, we first formalize the dual-target cross-domain recommendation problem. Then, we propose a novel Graphical and Attentional framework for Dual-Target Cross-Domain Recommendation, called GA-DTCDR. Finally, we present the detailed components of GA-DTCDR.

3.1 Problem Statement

First, for readability purposes, we list the important notations of this paper in Table 1. Then, we define the Dual-Target Cross-Domain Recommendation as follows.

Dual-Target Cross-Domain Recommendation (DTCDR). *Given two related domains a and b including explicit feedback (e.g., ratings and comments), implicit feedback (e.g.,*

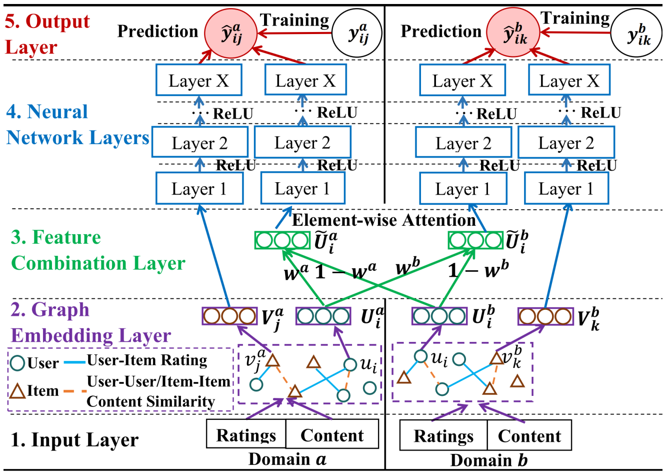


Figure 1: The overview of GA-DTCDR

purchase and browsing histories), and side information (e.g., user profiles and item details), the DTCDR is to improve the recommendation accuracies on both domains simultaneously by leveraging their observed information.

Note that a certain degree of overlap between the users of domains a and b, i.e., common users, plays a key role in bridging the two domains and exchanging knowledge across domains. This is a common idea in the existing single-target and dual-target CDR approaches.

3.2 Overview of GA-DTCDR

Targeting the dual-target CDR problem, we propose a novel graphical and attentional framework, called GA-DTCDR. As shown in Fig. 1, this framework is divided into five main components, i.e., *Input Layer*, *Graph Embedding Layer*, *Feature Combination Layer*, *Neural Network Layers*, and *Output Layer*. We will present the details of each component below.

Input Layer. First, for the input of our GA-DTCDR, we consider both explicit feedback (ratings and comments) and side information (user profiles and item details). These input data can be generally classified into two categories, i.e., rating information and content information.

Graph Embedding Layer. Then, we leverage the rating and content information of domains a and b to construct a heterogeneous graph representing user-item interaction relationships, user-user similarity relationships, and item-item similarity relationships. Based on the graph, we apply the Graph Embedding model, Node2vec [Grover and Leskovec, 2016], to generate user and item embedding matrices.

Feature Combination Layer. Next, we propose an element-wise attention mechanism to combine the common users' embeddings for domains a and b. This layer intelligently gives a set of weights to the two embeddings of a common user learned from both domains and generates a combined embedding for the common user, which remains his/her features learned from domains a and b with different proportions.

Neural Network Layers. In this component, we apply a fully-connected neural network, i.e., Multi-Layer Perceptrons (MLP), to represent a non-linear relationship between users and items on each domain.

Output Layer. Finally, we can generate final user-item interaction predictions. The training of our model is mainly based on the loss between predicted user-item interactions and observed user-item interactions.

In fact, like the single-target or dual-target CDR approaches in [Zhao *et al.*, 2017; Zhu *et al.*, 2018; Zhu *et al.*, 2019], our GA-DTCDR framework can be applied to Cross-System Recommendation (CSR) as well, where the two systems have the same domain but different users, and thus contain common items only, such as DoubanMovie and MovieLens (see Task 3 in **Experiments and Analysis**). Accordingly, in Fig. 1, we only need to replace common users with common items for supporting dual-target CSR.

3.3 Graph Embedding Layer

The existing embedding strategies for recommender systems mainly focus on representing the user-item interaction relationship. Apart from the user-item interaction relationship, we use a graph to represent user-user and item-item relationships as well. Therefore, based on the rating and content information observed from domains a and b, we construct two heterogeneous graphs, including nodes (users and items) and weighted edges (ratings and content similarities), for domains a and b, respectively. Then, we can generate more representative user and item embedding matrices. The Graph Embedding contains three main components, i.e., *Document Embedding*, *Graph Construction*, and *Output*.

Document Embedding. To construct the heterogeneous graph, we need to compute the content similarities between two users or two items. To this end, we consider multi-source content information, e.g., reviews, tags, user profiles, item details, observed from domains a and b, to generate user and item content embedding matrices. In this paper, we adopt the most widely used model, i.e., Doc2vec [Le and Mikolov, 2014], as the document embedding technique. The detailed document embedding process works as follows: (1) First, in the training set, for a user u_i, we collect the comments (reviews and tags) C_{i*} and the user profile up_i of u_i into the same document d_i, while for an item v_j, we collect the comments (reviews and tags) C_{*j} on the item and its item detail id_j into the same document d_{m+j}; (2) Next, we segment the words in the documents $D = \{d_1, d_2, ..., d_{m+n}\}$ by using the most widely used natural language tool, i.e., StanfordCoreNLP [Manning *et al.*, 2014]; (3) Finally, we apply Doc2vec model to map the documents D into the text vectors UC and VC for users and items, respectively.

Graph Construction. First, we link the users and items via their interaction relationships. The weights of these interaction edges are normalized ratings, i.e., $R/max(R)$. To consider the user-user and item-item relationships in the heterogeneous graph, we generate the synthetic edges between two users or two items according to their normalized content similarities (edge weights). The generation probability $P(i, l)$ of the edge between users u_i and u_l is as follows:

$$P(i,l) = \alpha \cdot sim(UC_i, UC_l), \tag{1}$$

where α is a hyper-parameter which controls the sampling probability and $sim(UC_i, UC_l)$ is the normalized cosine similarity between UC_i and UC_l. Similarly, we can obtain the generation probability between two items. Based on the user-item interaction relationships, user-user similarity relationships, and item-item similarity relationships, we can construct the heterogeneous graphs G^a and G^b for domains a and b, respectively.

Output. Based on the two heterogeneous graphs G^a and G^b, we employ the graph embedding model, i.e., Node2vec [Grover and Leskovec, 2016], to generate user embedding matrix U and item embedding matrix V.

3.4 Feature Combination Layer

In this layer, we combine the embeddings of common users learned from domains a and b by an element-wise attention mechanism. By this way, the combined embeddings of common users $\tilde{U}$ for each domain can remain both features learned from the two domains in different proportions. The traditional attention mechanism tends to select a certain part of representative features, and gives these features higher weights when generating the combined features [Bahdanau *et al.*, 2014]. Similarly, for a common user u_i, our element-wise attention mechanism tends to pay more attention to the more informative elements from each pair of elements in U_i^a and U_i^b. Thus our element-wise attention mechanism can generate two more representative embeddings $\tilde{U}_i^a$ and $\tilde{U}_i^b$ of the common user u_i for domains a and b, respectively. The structure of element-wise attention is shown in *Feature Combination Layer* of Fig. 1. The combined embedding $\tilde{U}_i^a$ of a common user u_i for domain a can be represented as:

$$\tilde{U}_i^a = W^a \odot U_i^a + (1 - W^a) \odot U_i^b, \tag{2}$$

where $\odot$ is the element-wise multiplication and $W^a \in \mathbb{R}^{m^a \times k}$ is the weight matrix for the attention network on domain a. Similarly, we can obtain the combined embedding $\tilde{U}_i^b$ of u_i for domain b.

Note that for the distinct users and all the items on domains a and b, we just reserve their embeddings without using the attention mechanism because they do not have dual embeddings on both domains a and b.

3.5 Training for Neural Network Layers and Output Layer

First of all, we train our model with the following objective function on domain a:

$$\min_{P^a, Q^a, \Theta^a} \sum_{y \in Y^{a+} \cup Y^{a-}} \ell(y, \hat{y}) + \lambda(\|P^a\|_F^2 + \|Q^a\|_F^2), \tag{3}$$

where $\ell(y, \hat{y})$ is a loss function between an observed interaction y and its corresponding predicted interaction $\hat{y}$ (see Eq. (5)), Y^{a+} and Y^{a-} denote all the observed and the unobserved user-item interactions on domain a respectively, $\|P^a\|_F^2 + \|Q^a\|_F^2$ is the regularizer (see Eq. (6)) and λ is a hyper-parameter which controls the importance of the regularizer. To avoid our model over-fitted to Y^+ (positive instances), we randomly select a certain number of unobserved user-item interactions as negative instances, denoted by $Y^-_{sampled}$, to replace Y^-. This training strategy has been widely used in the existing approaches [He *et al.*, 2017].

Datasets	Douban			MovieLens
Domains	Book	Music	Movie	Movie
#Users	2,110	1,672	2,712	10,000
#Items	6,777	5,567	34,893	9,395
#Interactions	96,041	69,709	1,278,401	1,462,905
Density	0.67%	0.75%	1.35%	1.56%

Tasks		**Sparser**	**Richer**	**Overlap**
CDR	**Task 1**	DoubanBook	DoubanMovie	#Common Users = 2,106
	Task 2	DoubanMusic	DoubanMovie	#Common Users = 1,666
CSR	**Task 3**	DoubanMovie	MovieLens	#Common Items = 4,115

Table 2: Experimental datasets and tasks

Based on rating information, the user-item interaction y_{ij} between a user u_i and an item v_i can be represented as:

$$y_{ij} = \begin{cases} r_{ij}, & \text{if } y_{ij} \in Y^+; \\ 0, & \text{if } y_{ij} \in Y^-_{sampled}; \\ null, & \text{otherwise.} \end{cases} \tag{4}$$

We choose a normalized cross-entropy loss which can be represented as:

$$\ell(y, \hat{y}) = \frac{y}{max(R)} \log \hat{y} + (1 - \frac{y}{max(R)}) \log(1 - \hat{y}), \tag{5}$$

where $max(R)$ is the maximum rating on a domain.

As shown in *Neural Network Layers* of Fig. 1, our GA-DTCDR employs a neural network, i.e., MLP, to represent a non-linear relationship between users and items. The input embedding matrices of users and items on domain a for the MLP are $P_{in}^a = [\tilde{U}^a; U^{ad}]$ and $Q_{in}^a = V^a$ respectively, where $\tilde{U}^a$ is the combined embedding matrix of common users for domain a, and U^{ad} is the embedding matrix of distinct users on domain a. Therefore the embedding of user u_i and item embedding of item v_j in the output layer of the MLP can be represented as:

$$\begin{aligned} P_i^a &= P_{out_i}^a = f(...f(f(P_{in_i}^a \cdot W_{P_1}^a) \cdot W_{P_2}^a)), \\ Q_j^a &= Q_{out_j}^a = f(...f(f(Q_{in_j}^a \cdot W_{Q_1}^a) \cdot W_{Q_2}^a)), \end{aligned} \tag{6}$$

where the activation function $f(*)$ is *ReLU*, $W_{P_1}^a, W_{P_2}^a$... and $W_{Q_1}^a, W_{Q_2}^a$... are the weights of multi-layer networks in different layers on domain a for $P_{in_i}^a$ and $Q_{in_j}^a$, respectively.

Finally, in *Output Layer* of Fig. 1, the predicted interaction $\hat{y}_{ij}$ between u_i and v_j on domain a is as follows:

$$\hat{y}_{ij}^a = cosine(P_i^a, Q_j^a) = \frac{P_i^a \cdot Q_j^a}{\|P_i^a\| \|Q_j^a\|}. \tag{7}$$

Similarly, we can obtain the predicted interaction $\hat{y}_{ij}^b$ on domain b.

4 Experiments and Analysis

We conduct extensive experiments on four real-world datasets to answer the following key questions: **Q1:** How does

Model			Training Data	Encoding	Embedding	Transfer Strategy
Baselines	Single-Domain Recommendation (SDR)	**NeuMF** [He *et al.*, 2017]	Rating	One-hot	Non-linear MLP	-
		DMF [Xue *et al.*, 2017]	Rating	Rating Vector	Non-linear MLP	-
	Single-Target Cross-Domain Recommendation (CDR)	**CTR-RBF** [Xin *et al.*, 2015]	Rating & Content	Topic Modeling	Linear MF	Mapping & Transfer Learning
		BPR_DCDCSR [Zhu *et al.*, 2018]	Rating	Random Initialization	Linear MF	Combination & MLP
		TMH [Hu *et al.*, 2019]	Rating & Content	One-hot	Non-linear MLP	Mapping & Transfer Learning & Attention
	Dual-Target CDR	**DMF_DTCDR_Concat** [Zhu *et al.*, 2019]	Rating & Content	Rating Vector	Non-linear MLP	Multi-task Learning & Concatenation
		DDTCDR [Li and Tuzhilin, 2019]	Rating	One-hot & Multi-hot	Non-linear MLP	Dual Transfer Learning
Our Methods	Dual-Target CDR	**GA-DTCDR_Average** (a variant of GA-DTCDR for ablation study)	Rating & Content	Heterogeneous Graph	Graph Embedding	Combination (**Average-Pooling**)
		GA-DTCDR	Rating & Content	Heterogeneous Graph	Graph Embedding	Combination (**Element-wise Attention**)

Table 3: The comparison of the baselines and our methods

Task	**Domain**	**SDR Baselines**				**Single-Target CDR Baselines**					
		NeuMF		DMF		CTR-RBF		BPR_DCDCSR		TMH	
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Task1 ($k = 8$)	DoubanBook (Sparser)	.3810	.2151	.3841	.2265	.3830	.2217	.3954	.2419	.4199	**.2583***
	DoubanMovie (Richer)	.5266	.2911	.5498	.3114	-	-	-	-	-	-
Task 1 ($k = 16$)	DoubanBook (Sparser)	.3833	.2181	.3854	.2356	.3870	.2256	.4014	.2413	.4331	**.2522***
	DoubanMovie (Richer)	.5282	.2939	.5573	.3141	-	-	-	-	-	-
Task 1 ($k = 32$)	DoubanBook (Sparser)	.3899	.2182	.3871	.2340	.3956	.2264	.4079	.2436	**.4468***	**.2647***
	DoubanMovie (Richer)	.5411	.2991	.5612	.3254	-	-	-	-	-	-
Task 1 ($k = 64$)	DoubanBook (Sparser)	.3908	.2226	.3917	.2362	.4017	.2314	.4107	.2454	**.4504***	**.2768***
	DoubanMovie (Richer)	.5449	.3152	.5632	.3387	-	-	-	-	-	-
Task 1 ($k = 128$)	DoubanBook (Sparser)	.4012	.2310	.4046	.2451	.4171	.2532	.4111	.2431	**.4523***	**.2814***
	DoubanMovie (Richer)	.5512	.3301	.5776	.3505	-	-	-	-	-	-

Task	**Domain**	**Dual-Target CDR Baselines**				**Dual-Target CDR (our)**				**Improvement (GA-DTCDR vs. best baselines)**	
		DMF_DTCDR _Concat		DDTCDR		GA-DTCDR _Average		**GA-DTCDR**			
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Task1 ($k = 8$)	DoubanBook (Sparser)	**.4412***	.2571	.4033	.2257	.4057	.2513	**.4479**	**.2759**	1.52%	6.81%
	DoubanMovie (Richer)	**.6032***	**.3732***	.5612	.3185	.5968	.3546	**.6518**	**.4025**	8.06%	7.85%
Task 1 ($k = 16$)	DoubanBook (Sparser)	**.4408***	.2513	.4054	.2292	.4190	.2577	**.4706**	**.2900**	6.76%	14.99%
	DoubanMovie (Richer)	**.6080***	**.3721***	.5750	.3595	.6013	.3596	**.6566**	**.4014**	10.80%	7.87%
Task 1 ($k = 32$)	DoubanBook (Sparser)	.4318	.2461	.4180	.2344	.4346	.2610	**.4758**	**.2896**	6.50%	9.41%
	DoubanMovie (Richer)	**.6011***	**.3718***	.5739	.3386	.6374	.3896	**.6747**	**.4187**	12.24%	12.61%
Task 1 ($k = 64$)	DoubanBook (Sparser)	.4265	.2452	.4258	.2430	.4423	.2671	**.4882**	**.3026**	8.40%	9.32%
	DoubanMovie (Richer)	**.5998***	**.3649***	.5825	.3553	.6416	.3941	**.6817**	**.4205**	13.65%	15.23%
Task 1 ($k = 128$)	DoubanBook (Sparser)	.4317	.2510	.4225	.2439	.4490	.2691	**.4995**	**.3098**	10.44%	10.09%
	DoubanMovie (Richer)	**.5991***	**.3680***	.5863	.3589	.6449	.3981	**.6957**	**.4406**	16.12%	19.73%

Table 4: The experimental results (HR@10 & NDCG@10) for Tasks 1 (the best-performing baselines with results marked by *)

our model outperform the state-of-the-art models (see Result 1)? **Q2:** How does the element-wise attention mechanism contribute to performance improvement (see Result 2)? **Q3:** How does the dimension k of embeddings affect the performance of our model (see Result 3)? **Q4:** How does our model perform on Top-N recommended lists (see Result 4)?

4.1 Experimental Settings

Experimental Datasets and Tasks. To validate the recommendation performance of our GA-DTCDR approach and baseline approaches, we choose four real-world datasets (see Table 2), i.e., three Douban subsets (DoubanBook, DoubanMusic, and DoubanMovie) [Zhu *et al.*, 2019], and MovieLens 20M [Harper and Konstan, 2016]. For the three Douban subsets, we retain the users and items with at least 5 interactions, while for MovieLens 20M, we extract a MovieLens subset containing $10,000$ users with at least 5 interactions as well. This filtering strategy has been widely used in the existing approaches [Yuan *et al.*, 2019; Zhu *et al.*, 2019]. The three Douban subsets contain ratings, reviews, tags, user profiles, and item details while MovieLens contains ratings, tags, and item details. Based on these four datasets, we design 2 CDR tasks and 1 CSR task (see Table 2) to validate the recommendation performance in CDR and CSR scenarios, respectively.

Parameter Setting. For fair comparison, we optimize the parameters of our GA-DTCDR and those of the baselines. For *Graph Embedding Layer* in Fig. 1, we set the hyper-parameters of Doc2vec and Node2vec models as suggested in [Le and Mikolov, 2014; Grover and Leskovec, 2016], and the sampling probability α as 0.05. In *Neural Network Layers* of Fig. 1, the structure of the layers is "$k \to 2k \to 4k \to 8k \to 4k \to 2k \to k$", the parameters of the neural network are initialized as the Gaussian distribution $X \sim \mathcal{N}(0, 0.01)$. For training our GA-DTCDR, we randomly select 7 negative instances for each observed positive instance into $Y^{-}_{sampled}$, adopt Adam [Kingma and Ba, 2014] to train the neural network, and set the maximum number of training epochs to 50. The learning rate is 0.001, the regularization coefficient λ is 0.001, and the batch size is $1,024$. To answer **Q3**, the dimension k of the embedding varies in $\{8, 16, 32, 64, 128\}$.

Task	**Domain**	**SDR Baselines**				**Single-Target CDR Baselines**					
		NeuMF		DMF		CTR-RBF		BPR_DCDCSR		TMH	
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Task 2	DoubanMusic (Sparser)	.3135	.1703	.3127	.1812	.3227	.1895	.3259	.1894	.3579	.2034
($k = 8$)	DoubanMovie (Richer)	.5266	.2911	.5498	.3114	-	-	-	-	-	-
Task 2	DoubanMusic (Sparser)	.3190	.1731	.3170	.1891	.3121	.1761	.3261	.1901	.3612	.2137
($k = 16$)	DoubanMovie (Richer)	.5282	.2939	.5573	.3141	-	-	-	-	-	-
Task 2	DoubanMusic (Sparser)	.3198	.1771	.3218	.1912	.3141	.1844	.3271	.1931	**.3701***	**.2202***
($k = 32$)	DoubanMovie (Richer)	.5411	.2991	.5612	.3254	-	-	-	-	-	-
Task 2	DoubanMusic (Sparser)	.3242	.1791	.3267	.1926	.3324	.1916	.3304	.2001	**.3882***	**.2323***
($k = 64$)	DoubanMovie (Richer)	.5449	.3152	.5632	.3387	-	-	-	-	-	-
Task 2	DoubanMusic (Sparser)	.3314	.1810	.3301	.1971	.3412	.1954	.3452	.2074	**.3946***	**.2430***
($k = 128$)	DoubanMovie (Richer)	.5512	.3301	.5776	.3505	-	-	-	-	-	-

Task	**Domain**	**Dual-Target CDR Baselines**				**Dual-Target CDR (our)**				**Improvement (GA-DTCDR vs. best baselines)**	
		DMF_DTCDR _Concat		DDTCDR		GA-DTCDR _Average		**GA-DTCDR**			
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Task 2	DoubanMusic (Sparser)	**.3614***	**.2117***	.3302	.1930	.3690	.2109	**.3852**	**.2166**	6.59%	2.31%
($k = 8$)	DoubanMovie (Richer)	**.5873***	**.3867***	.5655	.3629	.5987	.3731	**.6470**	**.3983**	10.17%	3.00%
Task 2	DoubanMusic (Sparser)	**.3663***	**.2213***	.3451	.2092	.3706	.2037	**.3947**	**.2256**	7.75%	1.94%
($k = 16$)	DoubanMovie (Richer)	**.5887***	**.3863***	.5704	.3676	.6058	.3716	**.6426**	**.3950**	9.16%	2.25%
Task 2	DoubanMusic (Sparser)	.3607	.2201	.3463	.2050	.3789	.2056	**.4133**	**.2318**	14.58%	5.32%
($k = 32$)	DoubanMovie (Richer)	**.5770***	**.3758***	.5739	.3726	.6145	.3754	**.6677**	**.4141**	15.72%	10.19%
Task 2	DoubanMusic (Sparser)	.3571	.2109	.3466	.2045	.3812	.2144	**.4384**	**.2489**	12.93%	7.15%
($k = 64$)	DoubanMovie (Richer)	**.5787***	**.3705***	.5719	.3621	.6120	.3681	**.6817**	**.4284**	17.80%	15.63%
Task 2	DoubanMusic (Sparser)	.3580	.2132	.3520	.2117	.3996	.2207	**.4491**	**.2604**	13.81%	7.16%
($k = 128$)	DoubanMovie (Richer)	**.5792***	.3742	.5748	**.3762***	.6311	.3859	**.7068**	**.4526**	22.03%	20.31%

Table 5: The experimental results (HR@10 & NDCG@10) for Tasks 2

Evaluation Metrics. To evaluate the recommendation performance of our GA-DTCDR approach and baseline approaches, we adopt the ranking-based evaluation strategy, i.e., *leave-one-out evaluation*, which has been widely used in the literature [Xue *et al.*, 2017; Wang *et al.*, 2019]. For each test user, we choose the latest interaction with a test item as the test interaction and randomly sample 99 unobserved interactions for the test user, and then rank the test item among the 100 items. *Leave-one-out evaluation* includes two main metrics, i.e., *Hit Ratio (HR)* and *Normalized Discounted Cumulative Gain (NDCG)* [Wang *et al.*, 2019]. *HR*@N is the recall rate while *NDCG*@N measures the specific ranking quality that assigns high scores to hits at top position ranks. Note that we only report *HR*@10 and *NDCG*@10 results in **Results 1-3**, and *HR*@N and *NDCG*@N results in **Result 4**.

Comparison Methods. As shown in Table 3, we compare our GA-DTCDR with seven baseline models in three groups, i.e., (1) Single-Domain Recommendation (SDR), (2) Single-Target Cross-Domain Recommendation (CDR), and (3) Dual-Target CDR. All of them are representative and/or state-of-the-art approaches for each group. Also, for ablation study, we also implement a simplified version of our GA-DTCDR, i.e., GA-DTCDR_Average (replacing element-wise attention with a fixed combination strategy, i.e., average-pooling). For clear comparison, in Table 3, we list the detailed training data types, encoding strategies, embedding strategies, and transfer strategies of all the nine models implemented in the experiments.

4.2 Performance Comparison and Analysis

Result 1: Performance Comparison (for Q1)

To answer **Q1**, we compare the performance of our GA-DTCDR with those of the seven baseline models. Note that for the SDR baselines, we train them on each domain and then report their performance on each domain; for the single-target CDR baselines, we train them on both domains and then only report their performance on the sparser domain; and for the dual-target CDR models, we train them on both domains and then report their performance on each domain.

Tables 4 - 6 show the experimental results in terms of HR@10 and NDCG@10 with different embedding dimensions k for Tasks 1, 2, and 3, respectively. As indicated in Tables 4 - 6, our GA-DTCDR outperforms all the SDR, single-target CDR, and dual-target CDR baselines by an average improvement of 8.46%. In particular, our GA-DTCDR improves the best-performing baselines (with results marked by * in Tables 4 - 6) by an average of 10.34% for Task 1, an average of 10.29% for Task 2, and an average of 4.76% for Task 3. This is because our GA-DTCDR effectively leverages the richness and diversity of the information on both domains, and intelligently and effectively combines the embeddings of common users.

Result 2: Ablation Study (for Q2)

To answer **Q2**, we implement a variant of our GA-DTCDR, i.e., GA-DTCDR_Average, by replacing element-wise attention with average-pooling. Average-pooling is the combination strategy used by the existing dual-target CDR approaches [Zhu *et al.*, 2019], which gives the weight equally, i.e., 0.5, to the embeddings of common users learned from dual domains. As we can see from Tables 4 - 6, with the element-wise attention, our GA-DTCDR improves GA-DTCDR_Average by an average of 6.76%. This means that element-wise attention plays a very important role in our GA-DTCDR and the existing fixed combination strategies can hardly achieve an effective embedding optimization on each target domain.

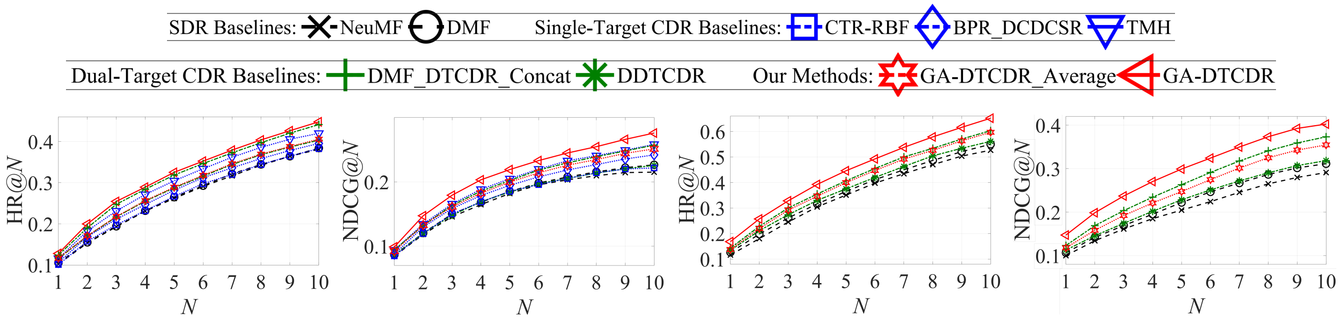
Result 3: Impact of Embedding Dimension k (for Q3)

To answer **Q3**, we analyze the effect of k on the performance of our GA-DTCDR as depicted in Tables 4 - 6. In general, in terms of HR@10 and NDCG@10, the recommendation accuracy of our GA-DTCDR increases with k because a larger

Task	**Domain**	**SDR Baselines**				**Single-Target CDR Baselines**					
		NeuMF		DMF		CTR-RBF		BPR_DCDCSR		TMH	
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Task 3 ($k = 8$)	DoubanMovie (Sparser)	.5266	.2911	.5498	.3114	.5514	.3156	.5762	.3347	.5987	.3487
	MovieLens (Richer)	.7818	.5024	.8115	.5219	-	-	-	-	-	-
Task 3 ($k = 16$)	DoubanMovie (Sparser)	.5282	.2939	.5573	.3141	.5631	.3213	.5816	.3438	.6031	.3580
	MovieLens (Richer)	.7901	.5084	.8143	.5212	-	-	-	-	-	-
Task 3 ($k = 32$)	DoubanMovie (Sparser)	.5411	.2991	.5612	.3254	.5721	.3347	.5821	.3447	.6108	**.3733***
	MovieLens (Richer)	.7978	.5124	.8180	**.5231***	-	-	-	-	-	-
Task 3 ($k = 64$)	DoubanMovie (Sparser)	.5449	.3152	.5632	.3387	.5704	.3327	.5926	.3559	.6186	**.3754***
	MovieLens (Richer)	.7935	.5149	**.8231***	.5277	-	-	-	-	-	-
Task 3 ($k = 128$)	DoubanMovie (Sparser)	.5512	.3301	.5776	.3505	.5912	.3741	.6142	.3904	.6314	**.3927***
	MovieLens (Richer)	.8042	.5205	**.8319***	.5344	-	-	-	-	-	-

Task	**Domain**	**Dual-Target CDR Baselines**				**Dual-Target CDR (our)**				**Improvement (GA-DTCDR vs. best baselines)**	
		DMF_DTCDR _Concat		DDTCDR		GA-DTCDR _Average		**GA-DTCDR**			
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Task 3 ($k = 8$)	DoubanMovie (Sparser)	**.6387***	**.3628***	.6070	.3522	.6140	.3572	**.6486**	**.4005**	6.85%	10.39%
	MovieLens (Richer)	**.8328***	**.5293***	.8211	.5283	.8225	.5241	**.8541**	**.5372**	2.56%	1.49%
Task 3 ($k = 16$)	DoubanMovie (Sparser)	**.6391***	**.3606***	.6100	.3518	.6266	.3710	**.6514**	**.4018**	1.92%	11.43%
	MovieLens (Richer)	**.8312***	**.5260***	.8263	.5170	.8280	.5277	**.8547**	**.5381**	2.83%	1.02%
Task 3 ($k = 32$)	DoubanMovie (Sparser)	**.6530***	.3631	.6137	.3460	.6310	.3776	**.6598**	**.4087**	1.04%	9.48%
	MovieLens (Richer)	**.8243***	.5213	.8111	.5167	.8301	.5280	**.8612**	**.5478**	4.48%	4.72%
Task 3 ($k = 64$)	DoubanMovie (Sparser)	**.6477***	.3605	.6200	.3544	.6423	.3841	**.6654**	**.4101**	2.73%	9.24%
	MovieLens (Richer)	.8200	**.5382***	.8130	.5198	.8324	.5320	**.8668**	**.5516**	5.31%	2.50%
Task 3 ($k = 128$)	DoubanMovie (Sparser)	**.6521***	.3642	.6222	.3714	.6489	.3792	**.6812**	**.4198**	4.46%	6.90%
	MovieLens (Richer)	.8267	**.5401***	.8210	.5311	.8349	.5381	**.8642**	**.5512**	3.88%	2.06%

Table 6: The experimental results (HR@10 & NDCG@10) for Tasks 3



(a) DoubanBook (HR@N) (b) DoubanBook (NDCG@N) (c) DoubanMovie (HR@N) (d) DoubanMovie (NDCG@N)

Figure 2: The result of Top-N recommendation for Task 1 ($k = 8$)

embedding can represent a user/item more accurately. However, considering the structure of the neural network layers in **Parameter Setting**, the training time of our GA-DTCDR also increases with k. This is a trade-off. Therefore, considering both aspects, $k = 64$ is ideal in our experiments.

Result 4: Top-N Recommendation Performance (for Q4)

To answer **Q4**, we compare the performance of top-N recommendation in terms of HR@N and NDCG@N where N ranges from 1 to 10. In fact, the performance trends of the all top-N experiments (for all the tasks with different k) are similar. So, due to space limitation, we only report the Top-N recommendation results of all the seven baseline models, GA-DTCDR_Average, and GA-DTCDR for Task 1 ($k = 8$) in Fig. 2. In Fig. 2, on both DoubanBook (sparser) and DoubanMovie (richer), the performance of our GA-DTCDR is consistently better than those of all the seven baselines and GA-DTCDR_Average. On DoubanBook, considering all the Top-N recommendations, our GA-DTCDR improves the best-performing baselines in different experimental cases by an average of 1.74% for HR@N, and by an average of 5.83% for NDCG@N, while on DoubanMovie, our GA-DTCDR improves the best-performing baselines in different experimental cases by an average of 8.13% for HR@N, and by an average of 7.55% for NDCG@N.

5 Conclusion and Future Work

In this paper, we have proposed a Graphical and Attention framework for Dual-Target Cross-Domain Recommendation, called GA-DTCDR, which proposes the graph embedding technique to generate more representative user and item embeddings and the element-wise attention mechanism to improve the recommendation accuracies on both domains simultaneously. Also, we have conducted extensive experiments to demonstrate the superior performance of our GA-DTCDR. In the future, we plan to extend our approach to multi-target recommendations and conduct more comprehensive experiments on new datasets to validate the impact of data sparsity and scale of common users on performance.

Acknowledgements

This work was supported by Australian Research Council Discovery Project DP180102378.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Berkovsky *et al.*, 2007] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Cross-domain mediation in collaborative filtering. In *ICUM*, pages 355–359, 2007.

[Chen *et al.*, 2017] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*, pages 335–344, 2017.

[Fu *et al.*, 2019] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *AAAI*, pages 94–101, 2019.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864, 2016.

[Harper and Konstan, 2016] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *TIIS*, 5(4):19, 2016.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.

[Hu *et al.*, 2019] Guangneng Hu, Yu Zhang, and Qiang Yang. Transfer meets hybrid: A synthetic approach for cross-domain collaborative filtering with text. In *WWW*, pages 2822–2829, 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.

[Li and Tuzhilin, 2019] Pan Li and Alexander Tuzhilin. D-dtcdr: Deep dual transfer cross domain recommendation. *arXiv preprint arXiv:1910.05189*, 2019.

[Liu *et al.*, 2019] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. Geniepath: Graph neural networks with adaptive receptive paths. In *AAAI*, pages 4424–4431, 2019.

[Lu *et al.*, 2018] Yichao Lu, Ruihai Dong, and Barry Smyth. Why i like it: multi-task learning for recommendation and explanation. In *RecSys*, pages 4–12, 2018.

[Man *et al.*, 2017] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI*, pages 2464–2470, 2017.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP toolkit. In *ACL System Demonstrations*, pages 55–60, 2014.

[Pan and Yang, 2009] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2009.

[Pan and Yang, 2013] Weike Pan and Qiang Yang. Transfer learning in heterogeneous collaborative filtering domains. *Artificial Intelligence*, 197:39–55, 2013.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[Tan *et al.*, 2014] Shulong Tan, Jiajun Bu, Xuzhen Qin, Chun Chen, and Deng Cai. Cross domain recommendation based on multi-type media fusion. *Neurocomputing*, 127:124–134, 2014.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. *arXiv preprint arXiv:1905.07854*, 2019.

[Wold *et al.*, 1987] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[Xin *et al.*, 2015] Xin Xin, Zhirun Liu, Chin-Yew Lin, Heyan Huang, Xiaochi Wei, and Ping Guo. Cross-domain collaborative filtering with review text. In *IJCAI*, pages 1827–1834, 2015.

[Xue *et al.*, 2017] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *IJCAI*, pages 3203–3209, 2017.

[Yuan *et al.*, 2019] Feng Yuan, Lina Yao, and Boualem Benatallah. Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. *arXiv preprint arXiv:1905.10760*, 2019.

[Zhang *et al.*, 2016] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. Multi-domain active learning for recommendation. In *AAAI*, pages 2358–2364, 2016.

[Zhao *et al.*, 2017] Lili Zhao, Sinno Jialin Pan, and Qiang Yang. A unified framework of active transfer learning for cross-system recommendation. *Artificial Intelligence*, 245:38–55, 2017.

[Zhou *et al.*, 2018] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

[Zhu *et al.*, 2018] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet A Orgun, and Jia Wu. A deep framework for cross-domain and cross-system recommendations. In *IJCAI*, pages 3711–3717, 2018.

[Zhu *et al.*, 2019] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. Dtcdr: A framework for dual-target cross-domain recommendation. In *CIKM*, pages 1533–1542. ACM, 2019.