DTCDR: A Framework for Dual-Target Cross-Domain Recommendation

Feng Zhu[1], Chaochao Chen[2], Yan Wang[1], Guanfeng Liu[1], and Xiaolin Zheng[3]
[1]Macquarie University, Sydney, Australia [2]Ant Financial Services Group, [3]Zhejiang University, Hangzhou, China
[1]feng.zhu3@hdr.mq.edu.au, {yan.wang, guanfeng.liu}@mq.edu.au [2]chaochao.ccc@antfin.com [3]xlzheng@zju.edu.cn

ABSTRACT

In order to address the data sparsity problem in recommender systems, in recent years, Cross-Domain Recommendation (CDR) leverages the relatively richer information from a source domain to improve the recommendation performance on a target domain with sparser information. However, each of the two domains may be relatively richer in certain types of information (e.g., ratings, reviews, user profiles, item details, and tags), and thus, if we can leverage such information well, it is possible to improve the recommendation performance on both domains simultaneously (i.e., dual-target CDR), rather than a single target domain only. To this end, in this paper, we propose a new framework, DTCDR, for Dual-Target Cross-Domain Recommendation. In DTCDR, we first extensively utilize rating and multi-source content information to generate rating and document embeddings of users and items. Then, based on Multi-Task Learning (MTL), we design an adaptable embedding-sharing strategy to combine and share the embeddings of common users across domains, with which DTCDR can improve the recommendation performance on both richer and sparser (i.e., dual-target) domains simultaneously. Extensive experiments conducted on real-world datasets demonstrate that DTCDR can significantly improve the recommendation accuracies on both richer and sparser domains and outperform the state-of-the-art single-domain and cross-domain approaches.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Collaborative filtering**; • **Computing methodologies → Multi-task learning**.

KEYWORDS

Recommender Systems; Cross-Domain Recommendation; Collaborative Filtering; Multi-task Learning

ACM Reference Format:

Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. DTCDR: A Framework for Dual-Target Cross-Domain Recommendation. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357992

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357992

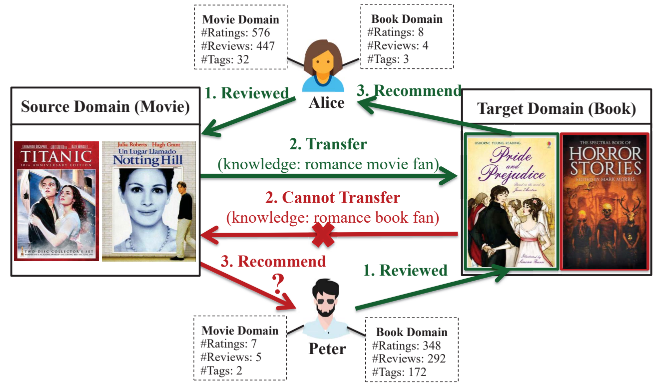


Figure 1: An example of a conventional single-target Cross-Domain Recommendation (CDR, Movie → Book).

1 INTRODUCTION

Collaborative Filtering (CF) has been proven to be one of the most promising techniques [32] in Recommender Systems (RSs). The main goal of CF techniques is to recommend items to a user based on the observed preferences of other users whose historical preferences are similar to those of the target user [14]. However, the existing CF-based RSs suffer from the long-standing data sparsity problem since few of the users can give ratings or reviews to many items [30]. This problem may lead to over-fitting when training a CF-based model, which significantly reduces recommendation accuracy.

A new trend for addressing this problem is to transfer relatively richer information from the source domain, e.g., observed ratings [9, 18, 20, 27, 44, 45], tags [1, 7], reviews [33], user/item attributes [5], and semantic networks [8], to improve the recommendation accuracy on the target domain with relatively sparser information, which is called Cross-Domain Recommendation (CDR) [5]. The conventional CDR approaches can be generally classified into two categories, i.e., content-based transfer and feature-based transfer. The general idea of *Content-Based Transfer* [5, 33] is to utilize different contents, e.g., reviews and user profiles, to match similar users or items, and build a bridge between the two domains, so that the similar users/items on different domains could share their information. The general idea of *Feature-Based Transfer* [20, 44, 45] is to first train different CF-based models, e.g., Probabilistic Matrix Factorization (PMF) [23] and Bayesian Personalized Ranking (BPR) [29], to obtain user/item features, and then transfer these features via common or similar users/items across domains.

All these existing CDR approaches only focus on how to leverage the source domain to help improve the recommendation accuracy on the target one, but not vice versa, namely, they are single-target CDR approaches. However, each of the two domains may be relatively richer in certain types of information (e.g., ratings, reviews, user profiles, item details, and tags), and thus, if we can leverage such information well, it is possible to improve the recommendation performance on both domains simultaneously, rather than a single target domain only. This is also explained in the following example.

A Motivating Example. Figure 1 depicts a conventional single-target cross-domain recommender system. It contains two domains, i.e., a *movie domain* with relatively richer comments and a *book domain* with sparser comments. Basically the conventional CDR approaches transfer knowledge learned from the source *movie domain* to improve the recommendation accuracy on the target *book domain*, but not vice versa. In a typical case, suppose Alice reviewed many movies, such as "Titanic" and "Notting Hill", but she reviewed a few books. Then, a conventional CDR system can recommend romance books, e.g., "Pride & Prejudice", rather than "Horror Stories", to Alice since she is a fan of romance movies. In a special case, suppose Peter reviewed many books and a few movies only, however, the conventional CDR system cannot make accurate recommendations of movies to him. This is because, in principle, the knowledge learned from the sparser domain is less accurate than that learned from the richer domain, which means the conventional CDR system cannot work well by simply changing the transfer direction as from the sparser domain to the richer domain. *Therefore*, in contrast to the conventional single-target CDR, it is necessary to devise a novel model to improve the recommendation accuracies for all users on both richer and sparser domains simultaneously by leveraging the data richness and diversity of both domains, i.e., *dual-target cross-domain recommendation (dual-target CDR)*.

Nevertheless, the novel dual-target CDR problem faces a new challenge without any solution reported in the literature, i.e., **CH1**: *'how to devise a feasible framework for dual-target CDR?'*. As an option, Multi-Task Learning (MTL) has the potential for dual-target CDR because it aims to improve models' generalization by leveraging the domain-specific information derived from the related recommendation tasks [31]. However, the existing MTL-based recommendation approaches [2, 19] cannot be efficiently applied to dual-target CDR. This is because they heavily rely on the local feature representation and side information (additional information associated with the users and items) from a single domain, and such features and information in the sparser domain may be too sparse to support dual-target CDR. In addition, Multi-Domain Recommendation (MDR) seems to be another option. However, the proposed MDR models in [25, 28, 41, 43] achieve different goals, i.e., they either focus on improving the recommendation accuracies of specific or common users selected from multiple domains, or only improve the recommendation accuracy on a single target domain. None of them can improve the recommendation accuracies of all users on multiple domains simultaneously. Therefore, the existing MDR models cannot serve for dual-target CDR directly.

Moreover, to address the data sparsity problem, multi-source information, such as ratings, reviews, user profiles, item details, and tags, derived from both domains should be leveraged to obtain more general user and item embeddings. Therefore, for dual-target CDR, there is another challenge **CH2**: *'how to effectively represent and combine these content and rating information to preserve the data diversity and richness of different domains and share them across domains to improve the recommendation accuracies on both domains simultaneously?'*.

Our Approach and Contributions: In this paper, we propose a new framework for dual-target CDR. To the best of our knowledge, this is the first work in the literature to propose the novel problem of dual-target CDR and provide a solution for it. The characteristics and contributions of our work are summarized as follows:

(1) Targeting **CH1**, we propose a novel framework for dual-target CDR, called DTCDR, which can leverage the data richness and diversity of dual domains, share the knowledge of common users across domains, and improve the recommendation accuracies for all users on both domains simultaneously;

(2) Targeting **CH2**, we first consider multi-source text information, including reviews, tags, user profiles, and item details, to generate the document embeddings of users and items by using Doc2Vec, and optimize two rating embedding models, i.e., NeuMF and DMF, to generate the rating embeddings of users and items. Then, based on MTL, we design an effective embedding-sharing strategy and choose three representative combination operators, i.e., Concatenation, Max-Pooling, and Average-Pooling, to respectively combine the text and rating embeddings of common users. They can synthesize these embeddings in diverse ways and make our framework adaptable to different scenarios;

(3) The extensive experiments conducted on real-world Douban and MovieLens datasets demonstrate that our approach significantly outperforms the state-of-the-art single-domain and cross-domain recommendation approaches in terms of recommendation accuracy.

2 RELATED WORK

In this section, we first review the related literature in two main categories in Sections 2.1 and 2.2: (1) Single-Domain Recommendation, and (2) Cross-Domain Recommendation. Because we employ multi-task learning for dual-target CDR, we also review the related literature about Multi-task Learning in Section 2.3.

2.1 Single-Domain Recommendation (SDR)

According to the focus of our work, we summarize the existing single-domain recommendation approaches in two groups, i.e., rating-based and content-based approaches.

Rating-Based: Conventional recommender systems have largely focused on making item recommendations based on observed ratings in a single domain. The rating-based approaches can be generally classified into two big categories according to different prediction techniques, i.e., Matrix Factorization (MF)-based approaches [3, 23, 29] and Neural Network (NN)-based approaches [12, 36, 38, 39]. The MF-based approaches tend to learn a linear relationship between users and items and their goals are to minimize the square or ranking loss between the observed and predicted ratings. Different from the MF-based approaches, the NN-based approaches apply a deep neural network, e.g., multi-layer perceptron (MLP), to learn a non-linear user-item interaction function and their goals are to minimize the loss between the observed and predicted interactions derived from ratings.

Content-Based: In addition, there are some content-based approaches [4, 10, 24] focussing on modeling both observed ratings and content information. Collaborative Topic Regression (CTR) model proposed in [35] is a big breakthrough for article (or citation) recommendation, which tends to combine the advantages of traditional collaborative filtering and topic modeling. The models in [4, 22] attempt to combine the latent factors learned from ratings with the latent review topics learned from contents by the topic models, which can explore more prior knowledge on users and items.

Summary: The existing single-domain recommendation approaches are constrained by the limited data from a single domain, which means that the data sparsity problem is hard to be further solved by these approaches.

2.2 Cross-Domain Recommendation (CDR)

According to transfer strategies, we review the existing CDR approaches into two categories, i.e., content-based transfer and feature-based transfer.

Content-Based Transfer: These approaches first create links by the common user/item attributes [5], semantic networks [8], social tags [1, 7, 15], and text information [33], then transfer user preferences or item details across domains.

Feature-Based Transfer: These approaches employ some classical machine learning models, e.g., multi-context learning [2], transfer learning [13, 26, 27, 44], and deep neural network [20, 45], to map or share user/item latent factors or rating patterns [18] learned by MF models across domains.

In addition, [25, 28, 41, 43] propose multi-domain models, but they either tend to make recommendations for specific or common users selected from domains or only for the users on the target domain. In contrast, our DTCDR framework aims to achieve the different goal, i.e., making recommendations for all users on both the source and target domains.

Summary: Compared with the single-domain recommendation approaches, the CDR approaches can leverage richer information from the source domain to improve the recommendation accuracy on the target domain. However, most of them are single-target approaches, which means they cannot leverage any information on the target domain to assist the source domain even if the target domain may be richer in certain types of data.

2.3 Multi-task Learning (MTL)

Multi-task Learning (MTL) is to synchronously learn multiple related tasks to take advantage of knowledge across the tasks, which is a good solution for addressing the data sparsity problem [42].

MTL Approaches: The MTL approaches can be broadly summarized into five categories [42], out of which, in this paper, we only focus on *Feature Learning* [6, 34, 40] for improving the performance of all the recommendation tasks by sharing a common embedding learned from all the tasks.

For Recommendation Tasks: The MTL-based approach proposed in [2] combines correlated context information from multiple tasks to improve predictive accuracy in recommender systems. In addition, the multi-task recommendation model proposed in [19] focuses on combining matrix factorization, for rating prediction, and adversarial sequence learning for recommendation explanation.

Summary: Although MTL has been applied for recommender systems, it is difficult for the existing MTL approaches to be efficiently applied for dual-target CDR because they heavily rely on the side information, i.e., additional information associated with the users and items, from a single domain, however, a sparser domain in dual-target CDR may be too sparse to support it.

Table 1: Important Notations

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
k	the dimension of embedding
m	the number of users
n	the number of items
P	the optimized embedding of users
Q	the optimized embedding of items
$r_{ij} \in R$	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 rating embedding of users
UC	the document embedding of users
$UP = \{up_1, ..., up_m\}$	the user profiles
$\mathcal{V} = \{v_1, ..., v_n\}$	the set of items
V	the rating embedding of items
VC	the document embedding 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 on domains A and B, e.g., $\mathcal{U}^a$ represents the set of users $\mathcal{U}$ on domain A
$\hat{*}$	the predicted notations, e.g., $\hat{Y}$ represents the predicted user-item interaction matrix

3 THE PROPOSED DTCDR FRAMEWORK

3.1 Notations and Problem Statement

For readability purposes, we list the important notations of this paper in Table 1. Based on these notations, we define the dual-target Cross-Domain Recommendation as follows:

Problem Formulation 1. ***Dual-Target Cross-Domain Recommendation:*** *Given two observed domains A and B which include the user ratings* $\{R^a, R^b\}$, *the user comments* $\{C^a, C^b\}$, *the user profiles* $\{UP^a, UP^b\}$, *and the item details* $\{ID^a, ID^b\}$, *the goal of dual-target CDR is to recommend the matched items* $\mathcal{V}_i$ *to any user* u_i *on any of the two domains, rather than only a user on the target domain for CDR.*

For dual-target CDR problem, a certain degree of overlap between the users of different domains, i.e., common users, is necessary, which can be used to link the two domains and share knowledge across them. This is common in the existing CDR approaches.

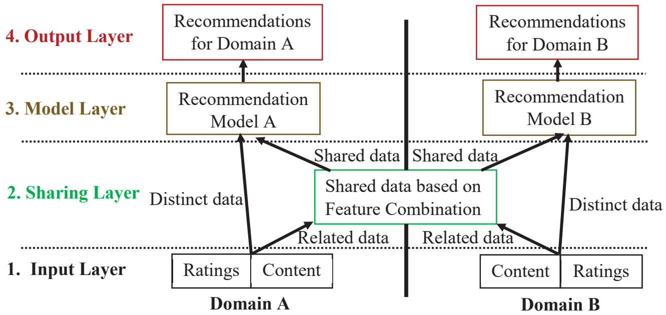


Figure 2: The general structure of DTCDR framework.

3.2 The General Framework for dual-target CDR

Targeting dual-target CDR, we propose a general framework, called DTCDR, as shown in Figure 2. The core idea of this framework is to utilize the data richness and diversity from the two domains A and B to improve the recommendation accuracies on both domains simultaneously. The framework contains four main parts, i.e., *Input Layer*, *Sharing Layer*, *Model Layer*, and *Output Layer*.

- **Input Layer**: First, as mentioned in Problem Formulation 1, the input of DTCDR framework contains users' explicit feedback (ratings and comments), user profiles, and item details. Different domains may be richer in certain types of input data. According to different methods of embedding processing, we divide the data of *Input Layer* into two types, i.e., ratings and content of users and items. *Input Layer* contains the input data from domains A and B.
- **Sharing Layer**: On the top of *Input Layer*, *Sharing Layer* mainly focuses on combining the related data of common users from two domains by combining features and sharing them for the recommendation models of the two domains. The main goal of feature combination is to utilize the data richness and diversity of common users from both domains.
- **Model Layer**: Next, in *Model Layer*, we take both the distinct data from a domain and the shared data from the two domains as the input, and train the recommendation model on each domain separately.
- **Output Layer**: Finally, the trained model can make recommendations for the corresponding domain in *Output Layer*.

Based on the core idea of our general DTCDR framework, we propose a specific Multi-task Learning-based solution in the following sections.

3.3 Multi-Task Learning-Based Solution for Our General DTCDR Framework

As shown in Figure 2, the recommendation models on different domains are parallel and closely related, thus the embeddings learned by ratings and content can be combined and shared by Multi-Task Learning (MTL). In this section, we propose a specific MTL-based solution for our DTCDR framework, as shown in Figure 3.

Based on the general framework in Figure 2, we further divide *Sharing Layer* into an *Embedding Layer* and a *Combination Layer*, and implement *Model Layer* with *Neural Network Layer*. Thus, the specific MTL-based solution (see Figure 3) includes five layers, i.e., *Input Layer*, *Embedding Layer*, *Combination Layer*, *Neural Network Layer*, and *Output Layer*.

- **Input Layer**: First, *Input Layer* contains the input data (ratings and content) from domains A and B.
- **Embedding Layer**: On the top of *Input Layer* is *Embedding Layer*, we generate document embedding UC for users and VC for items, and generate rating embedding U for users and V for items, respectively. The detailed embedding processes will be explained in Sections 3.4 and 3.5.
- **Combination Layer**: Next, On the top of *Embedding Layer* is *Combination Layer* which combines the document embedding and the rating embedding as the optimized embeddings P^a, Q^a, P^b, and Q^b, for users and items on the two domains. Specifically, we first choose *Max-Pooling* to combine the embeddings of common users learned from different domains because different domains are richer in certain types of input data and we expect to remain the dominating factors of them. Then, to be adaptable to different dual-target CDR scenarios, we design an effective embedding-sharing strategy, in which, we choose three representative combination operators, i.e., *Concatenation* (Concat), *Max-Pooling* (MP), and *Average-Pooling* (AP), to respectively combine the rating embedding and document embedding. Concat can preserve all embeddings learned from content and ratings, MP tends to remain their remarkable factors, and AP preserves the mean values of content and rating embeddings. These combination operators can utilize document and rating embeddings in diverse ways and make our models adaptable to different scenarios.
- **Neural Network Layer**: *Neural Network Layer* is used to model a non-linear interaction relationship between users and items, which can represent a complex user-item interaction relationship. We adopt a conventional multi-layer perceptron (MLP) in this layer.
- **Output Layer**: At last, based on *Neural Network Layer*, P, Q are mapped to the predicted user-item interaction matrix $\hat{Y}$ in *Output Layer*. The training process in this layer is to minimize the error between the predicted user-item interaction matrix $\hat{Y}$ and the observed user-item interaction matrix Y.

The specific MTL-based solution is presented in Algorithm 1 with details explained in the following sections.

In fact, our proposed DTCDR framework can apply to Cross-System Recommendation (CSR) [44, 45] 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 Section 4). Accordingly, in Figure 3, we need to replace common users with common items for supporting dual-target CSR.

3.4 Document Embedding for Embedding Layer

For the content information of each domain, we consider multi-source content information, e.g., reviews, user profiles, item details, and tags, of users and items to generate their text vectors by using document embedding. Document embedding is used to map documents or paragraphs to text vectors. The representative work for document embedding is Doc2Vec [17], which contains two

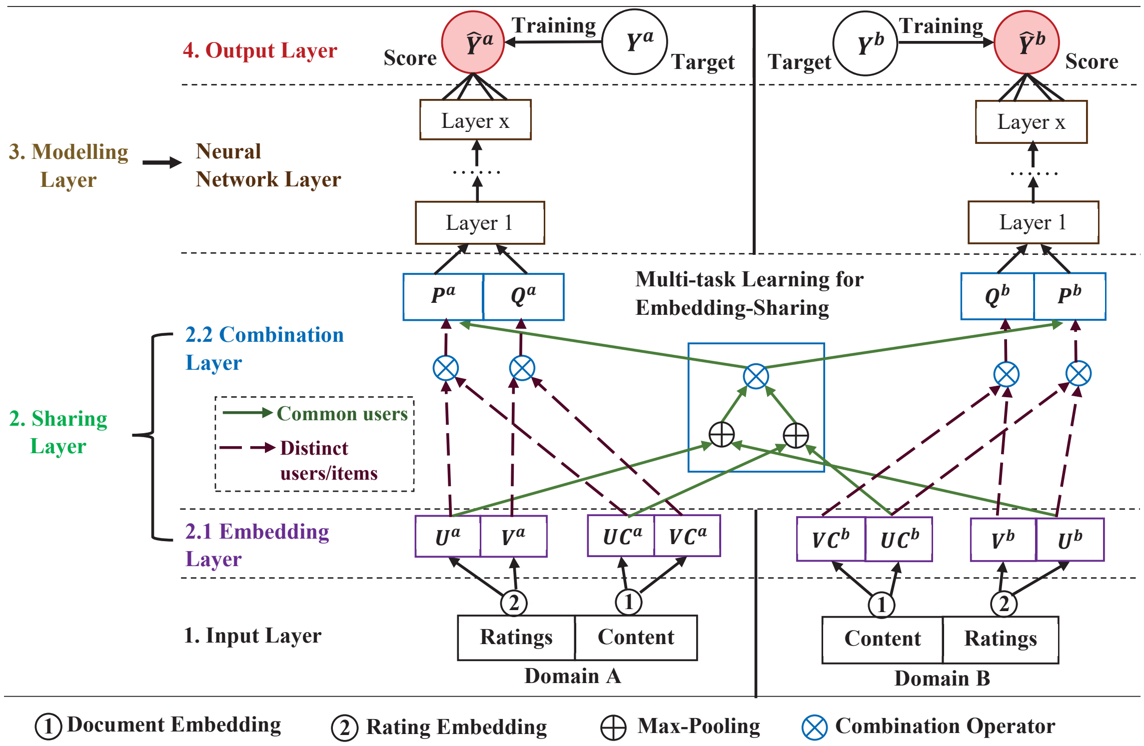


Figure 3: Our MTL-based solution for DTCDR framework.

sub-models, i.e., Distributed Memory (DM) and Distributed Bag Of Words (DBOW). We choose DBOW as the training algorithm because in our framework, the text vectors should not be affected by the word order in each document considering DBOW does not preserve any word order.

The detailed process works as follows: *First*, for user u_i, we combine u_i's user profile up_i with u_i's comments (reviews and tags) C_{i*} to get a document d_i, while for item v_j, we combine its item detail id_j with the comments C_{*j} on v_j to get a document d_{m+j}. *Then*, we apply the natural language tool StanfordCoreNLP [21] for cleaning text data and word segmentation of the documents $D = \{d_1, d_2, ..., d_{m+n}\}$. *Finally*, the documents D are mapped into corresponding text vectors UC and VC for users and items respectively by using the Doc2Vec model.

3.5 Rating Embedding for Embedding Layer

For the rating information of each domain, based on two popular neural network-based models, i.e., Neural Matrix Factorization (NeuMF) or Deep Matrix Factorization (DMF), we generate the latent factors U and V for users and items. In fact, in Section 3.6, we will optimize these two rating embedding models and make them deeply integrated with our framework. We briefly introduce NeuMF and DMF here, and the details can be found in [12, 38]. The general objective function can be represented as follows:

$$\min \sum_{y \in Y^+ \cup Y^-} \ell(y, \hat{y}) + \lambda \Omega(\Theta), \quad (1)$$

where $\ell(*)$ denotes a loss function, Y^+ denotes the observed user-item interactions, Y^- means all unobserved user-item interactions in Y, $\hat{y}$ is the predicted interactions for y, $\Omega(\Theta)$ is the regularizer and λ is a hyper-parameter which controls the importance of the regularizer. Note that we sample a certain number of negative instances, denoted by $Y^-_{sampled}$, to replace Y^-, which has been widely used in [12, 38].

NeuMF: NeuMF only considers implicit feedback deserved from explicit ratings and its user-item interaction can be represented as:

$$y_{ij} = \begin{cases} 1, \text{ if } r_{ij} \text{ is known;} \\ 0, \text{ otherwise.} \end{cases} \quad (2)$$

NeuMF employs a Generalized MF (GMF) model and a multilayer perceptron (MLP) to learn the user-item interaction function to predict ratings. NeuMF chooses the cross-entropy loss as its loss function, which can be formulated as follows:

$$\ell(y, \hat{y}) = y \log \hat{y} + (1 - y) \log(1 - \hat{y}). \quad (3)$$

DMF: Compared with NeuMF, DMF considers both implicit and explicit feedback and its user-item interaction can be represented as:

$$y_{ij} = \begin{cases} r_{ij}, \text{ if } r_{ij} \text{ is known;} \\ 0, \text{ otherwise.} \end{cases} \quad (4)$$

The core idea of DMF is to evaluate the cosine similarities between user and item latent factors learned by their own corresponding ratings. DMF improves the loss function of NeuMF by proposing a normalized cross entropy loss:

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

where $max(R)$ is the maximum rating in a dataset.

Algorithm 1 MTL-based solution for DTCDR framework

Require: (**Input Layer**) Two observed domains A and B, including the user ratings $\{R^a, R^b\}$, the user comments $\{C^a, C^b\}$, the user profiles $\{UP^a, UP^b\}$, the item details $\{ID^a, ID^b\}$, the number of training iterations num_iter, and the model type Mt (NeuMF_DTCDR or DMF_DTCDR).

Ensure: Recommend items $\mathcal{V}_i \subseteq \mathcal{V}$ to a target user u_i on any of the two domains.

1: # **Embedding Layer:**
2: Learn UC^a, VC^a from C^a, UP^a, and ID^a, by the document embedding model;
3: Pre-train U^a, V^a from R^a by rating embedding models;
4: Learn UC^b, VC^b from C^b, UP^b, and ID^b, by the document embedding model;
5: Pre-train U^b, V^b from R^b by rating embedding models;
6: **while** *epoch* from 1 to num_iter **do**
7:   # **Combination Layer:**
8:   Get the common users $\mathcal{U}^{ac} = \mathcal{U}^{bc} = \mathcal{U}^a \bigcap \mathcal{U}^b$;
9:   Get the distinct users $\mathcal{U}^{ad} = \mathcal{U}^a - \mathcal{U}^{ac}$ on domain A;
10:   Get the distinct users $\mathcal{U}^{bd} = \mathcal{U}^b - \mathcal{U}^{bc}$ on domain B;
11:   $U^c = U^{ac} \oplus U^{bc}$;
12:   $UC^c = UC^{ac} \oplus UC^{bc}$;
13:   $P^a = [U^c \otimes UC^c; U^{ad} \otimes UC^{ad}]$;
14:   $P^b = [U^c \otimes UC^c; U^{bd} \otimes UC^{bd}]$;
15:   $Q^a = V^a \otimes VC^a$;
16:   $Q^b = V^b \otimes VC^b$;
17:   # **Neural Network (NN) Layers & Output Layer:**
18:   Train the NN on domain A and B by Equation (6);
19:   **if** Mt is NeuMF_DTCDR **then**
20:     Predict user-item interactions $\hat{Y}^a$ on domain A by Equation (7);
21:   **else**
22:     Predict user-item interactions $\hat{Y}^a$ on domain A by Equation (8);
23:   **end if**
24:   Repeat Lines 19 to 23 to predict user-item interactions $\hat{Y}^b$ on domain B;
25: **end while**
26: **return** $\mathcal{V}_i$ according to $\hat{Y}^a$ and $\hat{Y}^b$.

3.6 Model Training

We train our DTCDR models by the following objective function on domain A:

$$\min_{P^a, Q^a, \Theta^a} \sum_{y \in Y^{a+} \cup Y^{a-}} \ell(y, \hat{y}) + \lambda(\|P^a\|_F^2 + \|Q^a\|_F^2),$$
$$[P^a, Q^a] = [[U^c \otimes UC^c; U^{ad} \otimes UC^{ad}], [V^a \otimes VC^a]], \tag{6}$$

where Θ^a is the parameter set for domain A, $\otimes$ is the combination operator, the predicted user-item interaction $\hat{y} \in \hat{Y}$ will be defined according to the specific rating embedding model in the following sections (see Equations (7) and (8)). In addition, U^c and UC^c represent the rating embedding and document embedding of common users from different domains while U^{ad}, UC^{ad}, V^a, and VC^a represent the embeddings of distinct users and all items from domain A. Likewise, we can obtain the objective function on domain B.

The detailed combination process for P^a, Q^a, P^b, and Q^b is shown between Lines 8 to 16 in Algorithm 1.

A DTCDR model through NeuMF (NeuMF_DTCDR): We first take NeuMF as the rating embedding model for our MLT-based solution. Thus, on domain A, the predicted user-item interaction of NeuMF_DTCDR for user u_i on item v_j can be defined as:

$$\hat{y}_{ij} = f\left(\begin{bmatrix} P_i^a \\ Q_j^a \end{bmatrix}\right),$$
$$P_i^a = \begin{cases} U_i \otimes UC_i, & u_i \in \mathcal{U}^{ac}; \\ U_i^a \otimes UC_i^a, & u_i \in \mathcal{U}^{ad}, \end{cases} \quad Q_j^a = V_j^a \otimes VC_j^a, \tag{7}$$
$$\begin{bmatrix} U_i^a \\ V_j^a \end{bmatrix} \leftarrow h^\top \begin{bmatrix} \phi_{ij}^{GMF^a} \\ \phi_{ij}^{MLP^a} \end{bmatrix}, \; h \leftarrow \begin{bmatrix} \alpha h^{GMF} \\ (1-\alpha) h^{MLP} \end{bmatrix},$$

where $f(*)$ is *ReLU* function, h^{GMF} and h^{MLP} denote the edge weights of the output layers of GMF and MLP models, respectively, and α is a hyper-parameter. In addition, $\mathcal{U}^{ac}$ and $\mathcal{U}^{ad}$ represent the common and distinct users, respectively, on domain A. Moreover, $\otimes$ represents a combination operator, i.e., Concat, MP, or AP. Here, we aim to provide flexibility and see which operator is more suitable in a special CDR scenario. Likewise, we can obtain the predicted user-item interactions of NeuMF_DTCDR on domain B.

A DTCDR model through DMF (DMF_DTCDR): We then take DMF as the rating embedding model for our MLT-based framework. Based on DMF model, we also choose the three different combination operators to combine the latent factors and the text vectors for users and items.

On domain A, the predicted user-item interaction of DMF_DTCDR for user u_i on item v_j can be redefined as:

$$\hat{y}_{ij} = cosine(P_i^a, Q_j^a) = \frac{[P_i^a]^\top Q_j^a}{\|Q_i^a\| \|Q_j^a\|},$$
$$U_i^a = f(\ldots f(W_{U_2}^a f(r_{i*} W_{U_1}^a))), \tag{8}$$
$$V_j^a = f(\ldots f(W_{V_2}^a f(r_{*j} W_{V_1}^a))),$$

where $f(*)$ is *ReLU* function, r_{i*} represents user u_i's ratings across all items, r_{*j} represents item v_j's ratings across all users, W_{U_1}, W_{U_2}... and W_{V_1}, W_{V_2}... are the weights of multi-layer networks in different layers for U and V, respectively. In addition, P_i^a and Q_j^a have been formulated in Equation (7). Likewise, we can obtain the predicted user-item interactions of DMF_DTCDR on domain B.

4 EXPERIMENTS AND ANALYSIS

We conduct extensive experiments on real-world datasets to answer the following five key questions:

- **Q1:** How does our approach outperform the state-of-the-art single-domain and cross-domain models (see Result 1)?
- **Q2:** How does the dimension k of latent factors and text vectors affect the performance of our models (see Result 2)?
- **Q3:** How do the three combination operators of MTL affect the performance of our models (see Result 3)?
- **Q4:** How do document embedding and MTL contribute to the performance improvement, respectively (see Result 4)?
- **Q5:** How does our approach perform on Top-N recommended lists (see Result 5)?

Table 2: Experimental datasets

Datasets	Douban			MovieLens
Domains	Music	Book	Movie	Movie
#Users	1,672	2,110	2,712	10,000
#Items	5,567	6,777	34,893	9,395
#Interactions	69,709	96,041	1,278,401	1,462,905
Sparsity	99.25%	99.33%	98.65%	98.44%

4.1 Experimental Settings

Datasets. In the experiments, we choose four real-world datasets, i.e., the benchmark dataset MovieLens 20M [11], three Douban datasets, including DoubanMusic, DoubanBook, and DoubanMovie, crawled from the Douban website[1]. We filter the three Douban datasets and keep the users and items with at least 5 interactions. In addition, for MovieLens 20M dataset, we choose $10,000$ users who have at least 5 interactions. The details are shown in Table 2. The three Douban datasets contain user profiles, item details, ratings, reviews, and tags while MovieLens dataset contains item genres, ratings, and tags. Based on these four datasets, we design three experimental tasks as follows.

Experimental Tasks. In order to validate the performance of our DTCDR models and baseline models in different CDR scenarios, we design two CDR tasks, i.e., Tasks 1 and 2. In addition, as mentioned in Section 3.3, our DTCDR models can apply to Cross-System Recommendation scenarios where there are common items only. Thus, we design Task 3 to validate the performance of our DTCDR and baseline models in a CSR scenario. The detailed tasks are listed as follows:

- **Task 1:** DoubanMovie+DoubanBook ($2,106$ common users)
- **Task 2:** DoubanMovie+DoubanMusic ($1,666$ common users)
- **Task 3:** DoubanMovie+MovieLens ($4,115$ common movies)

Parameter Setting. For a fair comparison, we optimized both the parameters of our DTCDR models and those of the baseline models. For *Input Layer* and *Embedding Layer* in Figure 3, we set the hyper-parameters of Doc2Vec model as suggested in [17] and the dimension k of the text vectors and latent factors as $\{8, 16, 32, 64\}$. In *Neural Network Layer*, the structure is $e \rightarrow 32 \rightarrow 16 \rightarrow k$, where k is the output size, i.e., the dimension of the latent factors, and e is the combined size. For different rating embedding models and combination operators, e has different values. For example, for DMF_DTCDR, if the combination operator is Concat, $e = 2 * k$; elsewise, $e = k$. The parameters of the neural network are initialized as the Gaussian distribution $X \sim N(0, 0.01)$. For NeuMF_DTCDR, $\lambda = 0.001$, the learning rate is 0.001 and the batch size is $1,024$, while for DMF_DTCDR, we set λ as 0.001, the learning rate as 0.0001 and the batch size as 256. We adopt the Adaptive Moment Estimation (Adam) [16] in our models. In addition, we set the number of training iterations *num_iter* as 50 and report the best performance in our experimental results.

Evaluation Metrics. We adopt the ranking-based evaluation strategy, i.e., *leave-one-out evaluation*, which has been widely used in the baseline models, e.g., BPR, NeuMF, and DMF. That is, for each test rating from a test user on a test item, we randomly sample 99 unrated items for the test user and then rank the test item among the 100 items. The recommendation performance is evaluated by the two metrics, i.e., *Hit Ratio (HR)* and *Normalized Discounted Cumulative Gain (NDCG)* [12]. *HR* measures whether the test item is ranked on the top-N list while *NDCG* measures the specific ranking quality that assigns high scores to hits at top position ranks.

Comparison Methods. We compare our NeuMF_DTCDR and DMF_DTCDR models with the following seven baseline models in two groups SDR and CDR, respectively. These baseline models are the most relevant methods because each of them is a representative or state-of-the-art method with different embedding and transfer strategies. For a clear comparison, we list the detailed training data types, embedding strategies and transfer strategies of the baseline models and our DTCDR models in Table 3.

(1) Single-Domain Recommendation (SDR baselines)

- *Bayesian Personalized Ranking* (**BPR**) [29] is a representative pairwise learning-based MF model, focusing on minimizing the ranking loss between predicted ratings and observed ratings.
- *Neural Matrix Factorization* (**NeuMF**) [12] is a representative NN-based CF model, replacing the conventional inner product with a neural architecture to improve recommendation accuracy.
- *Deep Matrix Factorization* (**DMF**) [38] is the state-of-the-art NN-based CF model, employing a deep architecture to learn the low-dimensional factors of users and items.

(2) Cross-Domain Recommendation (CDR baselines)

- *A Non-Linear Transfer Learning Framework* (**CTR-RBF**) [37] is a non-linear transfer learning framework with the review text incorporated. This is the state-of-the-art CDR model considering both content and rating information to generate user and item embeddings.
- *Embedding and Mapping framework* (EMCDR) [20] utilizes Linear Matrix Translation (LIN) and Multi-Layer Perceptron (MLP) to represent the relations between the latent factors of two domains. We implement the most promising models **BPR_EMCDR_LIN** and **BPR_EMCDR_MLP** in this framework as baselines.
- *Deep CDR framework* (DCDCSR) [45] is the state-of-the-art deep framework to transfer latent factors across domains. We implement the most promising model **BPR_DCDCSR** as a baseline in our experiments.

4.2 Performance Comparison and Analysis

To answer the five key questions **Q1-Q5**, we conduct the following experiments and analyze the corresponding results.

Result 1: Performance Comparison (for Q1)

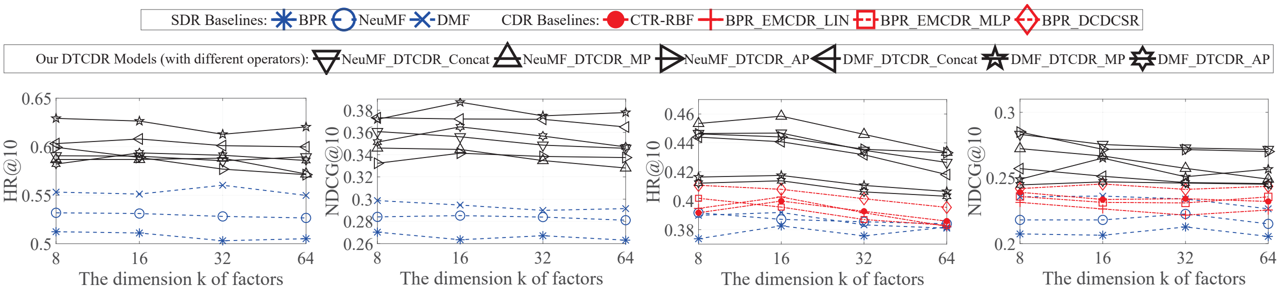
In order to answer **Q1**, we compare the performance of our proposed NeuMF_DTCDR and DMF_DTCDR models, with that of the seven baseline models. For the three single-domain recommendation models, we train them on both domains and obtain the corresponding experimental results. For the four cross-domain baseline models, we train them based on the two domains and validate them on the target (sparser) domain. The target domains are DoubanBook, DoubanMusic, and DoubanMovie for Tasks 1, 2, and 3, respectively.

Figures 4, 5 and 6 show the performance of HR@10 and NDCG@10 with different factor dimensions for Tasks 1, 2, and 3, respectively. On both the sparser and richer domains, on average, our NeuMF

[1] https://www.douban.com, a popular Chinese review site.

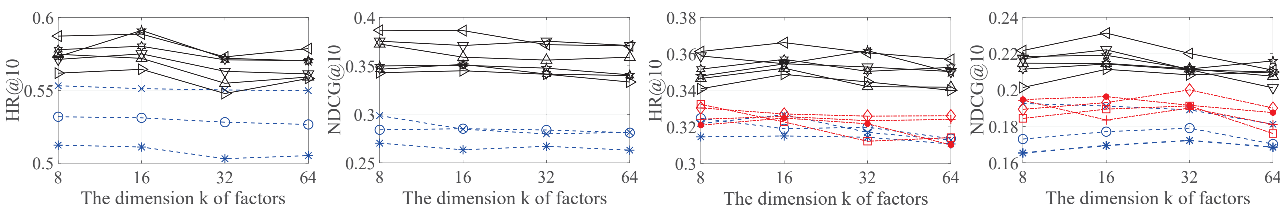
Table 3: The comparison of the baselines and our models

Model			Training Data	Embedding Strategy	Transfer Strategy (for single-target CDR or dual-target CDR)
Baselines	Single-Domain Recommendation (SDR)	BPR [29]	Rating	MF	-
		NeuMF [12]	Rating	MLP	-
		DMF [38]	Rating	MLP	-
	Single-target Cross-Domain Recommendation (CDR)	CTR-RBF [37]	Rating & Content	MF	Transfer Learning
		BPR_EMCDR_LIN [20]	Rating	MF	Linear Matrix Translation
		BPR_EMCDR_MLP [20]	Rating	MF	MLP
		BPR_DCDCSR [45]	Rating	MF	Feature Combination & MLP
Our DTCDR models	NeuMF-based	NeuMF_DTCDR_Concat	Rating & Content	MLP	MTL & **Concatenation**
		NeuMF_DTCDR_MP	Rating & Content	MLP	MTL & **Max-Pooling**
		NeuMF_DTCDR_AP	Rating & Content	MLP	MTL & **Average-Pooling**
	DMF-based	DMF_DTCDR_Concat	Rating & Content	MLP	MTL & **Concatenation**
		DMF_DTCDR_MP	Rating & Content	MLP	MTL & **Max-Pooling**
		DMF_DTCDR_AP	Rating & Content	MLP	MTL & **Average-Pooling**



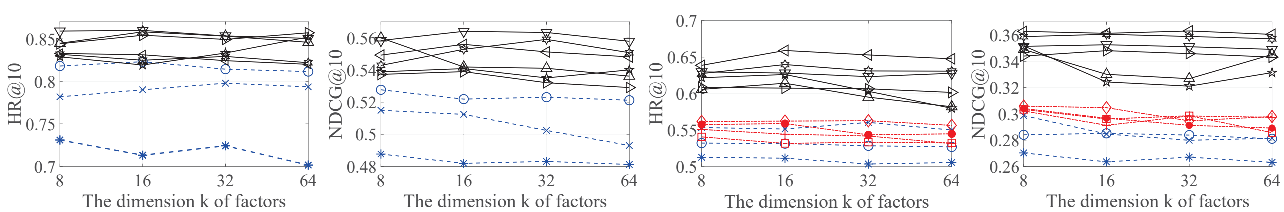
(a) DoubanMovie (HR@10) (b) DoubanMovie (NDCG@10) (c) DoubanBook (HR@10) (d) DoubanBook (NDCG@10)

Figure 4: The experimental result of Task 1. Note: DoubanBook is the target domain for CDR baseline models.



(a) DoubanMovie (HR@10) (b) DoubanMovie (NDCG@10) (c) DoubanMusic (HR@10) (d) DoubanMusic (NDCG@10)

Figure 5: The experimental result of Task 2. Note: DoubanMusic is the target domain for CDR baseline models.



(a) MovieLens (HR@10) (b) MovieLens (NDCG@10) (c) DoubanMovie (HR@10) (d) DoubanMovie (NDCG@10)

Figure 6: The experimental result of Task 3. Note: DoubanMovie is the target domain for CDR baseline models.

_DTCDR and DMF_DTCDR models outperform the three single-domain baseline models, i.e., BPR, NeuMF, and DMF, by 13.73%, 10.82%, and 8.97% respectively for HR@10, and by 18.83%, 13.98%, and 11.77% respectively for NDCG@10. On the sparser (target) domains, on average, our NeuMF_DTCDR and DMF_DTCDR models outperform the four cross-domain baseline models, i.e., CTR-RBF, BPR_EMCDR_LIN, BPR_EMCDR_MLP, and BPR_DCDCSR, by 9.63%, 9.90%, 10.06%, and 6.20% respectively for HR@10, and by 11.91%, 13.40%, 13.84%, and 9.37% respectively for NDCG@10. It is worth noting that in most cases (46 out of 48 cases), on average, our worst-performing model still outperforms the best-performing baseline model by 3.09% for HR@10 and 5.88% for NDCG@10, respectively. Meanwhile, in all the cases, on average, our best-performing

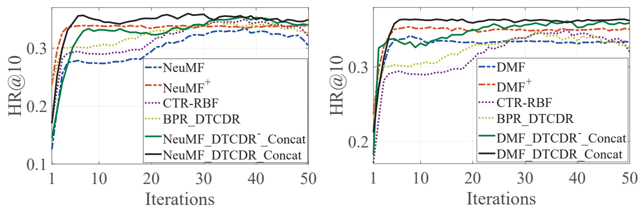


Figure 7: Performance comparison with and without document embedding (DE) on DoubanMusic (k = 8 and the combination operator is Concat). Note that NeuMF$^+$ and DMF$^+$ represent NeuMF and DMF with DE while NeuMF_DTCDR$^-$ and DMF_DTCDR$^-$ represent NeuMF_DTCDR and DMF_DTCDR without DE.

model improves the best-performing baseline model by 9.45% for HR@10 and 13.90% for NDCG@10, respectively.

Summary 1: In general, our NeuMF_DTCDR and DMF_DTCDR models outperform both the single-domain baseline models and the cross-domain baseline models. This is because our models can leverage the richness and diversity of the information of both domains and effectively share the embeddings of common users across domains, avoiding over-fitting. Our models can improve the recommendation performance on the two domains or systems, which illustrates the effectiveness of our DTCDR models.

Result 2: Impact of Factor Dimension (for Q2)

To answer **Q2**, we analyze the effect of k on model performance in Figures 4, 5, and 6; specifically, when $k \in \{8, 16\}$, our models can achieve the best performance on the three experimental tasks. When $k \in \{32, 64\}$, the performance of our models starts to decline gradually. This is because the number of parameters of neural network geometrically increases with k while the training data is relatively sparser, which can lead to over-fitting when $k > 16$.

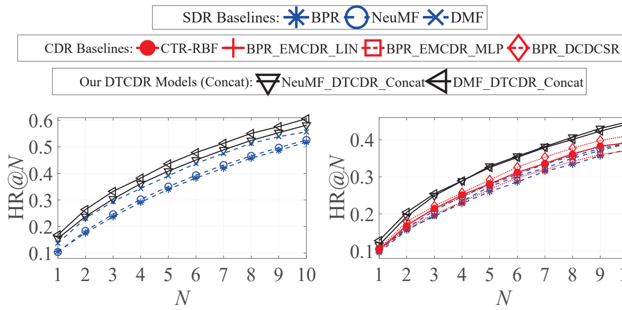
Summary 2: The dimension k of factors is a sensitive parameter in our NeuMF_DTCDR and DMF_DTCDR models. In general, when $k \leq 16$, the recommendation performance of our NeuMF_DTCDR and DMF_DTCDR models increases with k. However, when $k > 16$, the performance starts to decline gradually due to over-fitting.

Result 3: Impact of Combination Operators (for Q3)

In order to answer **Q3**, we compare the performance of our NeuMF_DTCDR and DMF_DTCDR models which have different combination operators, i.e., Concatenation (Concat), Max-Pooling (MP), and Average-Pooling (AP). From Figures 4, 5, and 6, when compared with *MP* and *AP*, we can see with *Concat*, NeuMF_DTCDR and DMF_DTCDR models can achieve the best performance in most cases. This is because, to a large extent, *Concat* can preserve all the embeddings of different domains. Meanwhile, *MP* tends to preserve dominating factors but lose generality while *AP* preserves generality but is easily affected by noisy embeddings.

Summary 3: *Concat* can bring better and stable performance for our NeuMF_DTCDR and DMF_DTCDR models. In some isolated cases, *MP* can achieve good performance only in quite a few cases. The performance of *AP* is worse than the these two operators mainly because *AP* can be easily affected by noisy embeddings.

Result 4: Contributions of document embedding and multi-task learning (for Q4)



(a) Douban-Movie (HR@10) (b) Douban-Book (HR@10)

Figure 8: The result of Top-N recommendation for Task 1.

We compare the performance of NeuMF, DMF, NeuMF_DTCDR, DMF_DTCDR with and without document embedding, CTR-RBF, and BPR_DCDCSR. This experiment is conducted on DoubanMusic dataset and evaluated by HR@10 with k = 8. As we can see from Figure 7, *with the assistance of document embedding*, the performance of NeuMF$^+$ and DMF$^+$ is always better than their pure models, i.e., NeuMF and DMF. According to the best performance among all 50 iterations, their improvements are 1.50% and 2.34%, respectively. Meanwhile, NeuMF_DTCDR and DMF_DTCDR outperform their simplified models (without document embedding), i.e., NeuMF_DTCDR$^-$ and DMF_DTCDR$^-$, by 1.84% and 1.07%, respectively.

In addition, *without document embedding, and only based on MTL*, our NeuMF_DTCDR$^-$ still outperforms NeuMF, NeuMF$^+$, CTR-RBF, and BPR_DCDCSR after 27 iterations, and our DMF_DTCDR$^-$ still outperforms DMF, DMF$^+$, CTR-RBF, and BPR_DCDCSR after 20 iterations. According to the best performance of all the models among all 50 iterations, our models NeuMF_DTCDR$^-$ and DMF_DTCDR$^-$ outperform all the six baseline models, i.e., NeuMF, NeuMF$^+$, DMF, DMF$^+$, CTR-RBF, and BPR_DCDCSR, by 6.12%, 4.79%, 5.72%, 4.38%, 3.35%, and 2.07%, respectively.

Summary 4: Document embedding can improve recommendation performance since the text vectors learned by document embedding can provide more prior knowledge to the recommendation models instead of the only initialization by a random or Gaussian distribution. In addition, without document embedding, our NeuMF-DTCDR$^-$ and DMF-DTCDR$^-$ can still achieve good performance as long as the models can be well trained. This is because Multi-task Learning technique is applied for sharing the features of common users and items across domains, which can effectively mitigate the data sparsity problem.

Result 5: Performance of Top-N Recommendation (for Q5)

In order to answer **Q5**, we compare the performance of Top-N recommendation in terms of HR@N where the ranking position N ranges from 1 to 10 and k is 8. To clearly show the comparison of the performance, we only report the performance of NeuMF_DTCDR_Concat, DMF_DTCDR_Concat and the baseline models for Task 1. As we can see from Figure 8, on both datasets, the performance of our NeuMF_DTCDR_Concat and DMF_DTCDR _Concat is consistently better than that of the other seven baseline models. On Douban-Movie, DMF_DTCDR_Concat has better performance than NeuMF_DTCDR_Concat while on Douban-Book, the performance of NeuMF_DTCDR_Concat is better that of DMF

_DTCDR_Concat when N is greater than 4. In this case, on average, our NeuMF_DTCDR_Concat and DMF_DTCDR_Concat improve the single-domain baselines, i.e., BPR, NeuMF, and DMF, by 18.36%, 15.62%, and 12.24% respectively, and improve the cross-domain baselines, i.e., CTR-RBF, BPR_EMCDR_LIN, BPR_EMCDR_MLP, and BPR_DCDCSR, by 12.16%, 12.20%, 16.71%, and 8.55% respectively.
Summary 5: In general, our NeuMF_DTCDR_Concat and DMF_DTCDR_Concat outperform the seven baseline models for Top-N recommendation, and DMF_DTCDR_Concat have better performance than NeuMF_DTCDR_Concat in most cases. This is because our models can leverage the richness and diversity of the information of both domains and effectively share the embeddings of common users across domains.

5 CONCLUSION AND FUTURE WORK

In this paper, we have proposed a general framework for Dual-Target Cross-Domain Recommendation, called DTCDR, which leverages ratings and multi-source content to improve the recommendation performance on dual-target domains simultaneously. We optimized document embedding and rating embedding techniques to generate the text and rating embeddings of users and items. Based on multi-task learning, we adopt a flexible and effective embedding-sharing strategy to combine and share the embeddings of common users across domains. Finally, extensive experiments conducted on real-world datasets have demonstrated the superior performance of our models.

In our future work, we plan to extend our approach to multi-target recommendations. In addition, we will study the impact of the proportion of common users between multiple domains and the sparsity of datasets on recommendation performance.

ACKNOWLEDGEMENTS

This work is partially supported by Australian Research Council Discovery Project DP180102378.

REFERENCES

[1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011. Analyzing user modeling on twitter for personalized news recommendations. *UMAP* (2011), 1–12.
[2] Deepak Agarwal, Bee-Chung Chen, and Bo Long. 2011. Localized factor models for multi-context recommendation. In *SIGKDD*. 609–617.
[3] Natalie Aizenberg, Yehuda Koren, and Oren Somekh. 2012. Build your own music recommender by modeling internet radio streams. In *WWW*. 1–10.
[4] Yang Bao, Hui Fang, and Jie Zhang. 2014. TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation. In *AAAI*, Vol. 14. 2–8.
[5] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-domain mediation in collaborative filtering. In *UM*. Springer, 355–359.
[6] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
[7] Ignacio Fernández-Tobías and Iván Cantador. 2014. Exploiting Social Tags in Matrix Factorization Models for Cross-domain Collaborative Filtering. In *RecSys*. 34–41.
[8] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2011. A generic semantic-based framework for cross-domain recommendation. In *HetRec*. 25–32.
[9] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. 2013. Cross-domain recommendation via cluster-level latent factor model. In *ECML and PKDD*. 161–176.
[10] Nur Al Hasan Haldar, Jianxin Li, Mark Reynolds, Timos Sellis, and Jeffrey Xu Yu. 2019. Location prediction in large-scale social networks: an in-depth benchmarking study. *The VLDB Journal* (2019), 1–26.
[11] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *TIIS* 5, 4 (2016), 19.
[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.
[13] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *CIKM*. ACM, 667–676.
[14] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender systems: an introduction*.
[15] Marius Kaminskas and Francesco Ricci. 2011. Location-adapted music recommendation using tags. *UMAP* (2011), 183–194.
[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[17] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. 1188–1196.
[18] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction. In *IJCAI*. 2052–2057.
[19] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like it: multi-task learning for recommendation and explanation. In *RecSys*. 4–12.
[20] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *IJCAI*. 2464–2470.
[21] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL System Demonstrations*. 55–60.
[22] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. 165–172.
[23] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NIPS*. 1257–1264.
[24] Raymond J Mooney and Loriene Roy. 2000. Content-based book recommending using learning for text categorization. In *DL*. 195–204.
[25] Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. 2012. Talmud: transfer learning for multiple domains. In *CIKM*. ACM, 425–434.
[26] Weike Pan, Nathan N Liu, Evan W Xiang, and Qiang Yang. 2011. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI*. 2318.
[27] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *AAAI*. 230–235.
[28] Weike Pan and Qiang Yang. 2013. Transfer learning in heterogeneous collaborative filtering domains. *Artificial Intelligence* 197 (2013), 39–55.
[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.
[30] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender Systems Handbook* (2nd ed.).
[31] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).
[32] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.
[33] Shulong Tan, Jiajun Bu, Xuzhen Qin, Chun Chen, and Deng Cai. 2014. Cross domain recommendation based on multi-type media fusion. *Neurocomputing* 127 (2014), 124–134.
[34] Michalis K Titsias and Miguel Lázaro-Gredilla. 2011. Spike and slab variational inference for multi-task and multiple kernel learning. In *NIPS*. 2339–2347.
[35] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *SIGKDD*. 448–456.
[36] Huizhao Wang, Guanfeng Liu, An Liu, Zhixu Li, and Kai Zheng. 2019. DMRAN:A Hierarchical Fine-Grained Attention-Based Network for Recommendation. In *IJCAI*. 3698–3704.
[37] Xin Xin, Zhirun Liu, Chin-Yew Lin, Heyan Huang, Xiaochi Wei, and Ping Guo. 2015. Cross-Domain Collaborative Filtering with Review Text. In *IJCAI*. 1827–1834.
[38] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *IJCAI*. 3203–3209.
[39] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI*. 4320–4326.
[40] Wenlu Zhang, Rongjian Li, Tao Zeng, Qian Sun, Sudhir Kumar, Jieping Ye, and Shuiwang Ji. 2016. Deep model based transfer and multi-task learning for biological image analysis. *IEEE transactions on Big Data* (2016).
[41] Yu Zhang, Bin Cao, and Dit-Yan Yeung. 2012. Multi-domain collaborative filtering. *arXiv preprint arXiv:1203.3535* (2012).
[42] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).
[43] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. 2016. Multi-Domain Active Learning for Recommendation. In *AAAI*. 2358–2364.
[44] Lili Zhao, Sinno Jialin Pan, and Qiang Yang. 2017. A unified framework of active transfer learning for cross-system recommendation. *Artificial Intelligence* 245 (2017), 38–55.
[45] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet A Orgun, and Jia Wu. 2018. A Deep Framework for Cross-Domain and Cross-System Recommendations. In *IJCAI*. 3711–3717.