Heterogeneous Multidomain Recommender System Through Adversarial Learning

Wenhui Liao, Qian Zhang, *Member, IEEE*, Bo Yuan, *Senior Member, IEEE*, Guangquan Zhang, and Jie Lu, *Fellow, IEEE*

***Abstract*— To solve the user data sparsity problem, which is the main issue in generating user preference prediction, cross-domain recommender systems transfer knowledge from one source domain with dense data to assist recommendation tasks in the target domain with sparse data. However, data are usually sparsely scattered in multiple possible source domains, and in each domain (source/target) the data may be heterogeneous, thus it is difficult for existing cross-domain recommender systems to find one source domain with dense data from multiple domains. In this way, they fail to deal with data sparsity problems in the target domain and cannot provide an accurate recommendation. In this article, we propose a novel multidomain recommender system (called HMRec) to deal with two challenging issues: 1) how to exploit valuable information from multiple source domains when no single source domain is sufficient and 2) how to ensure positive transfer from heterogeneous data in source domains with different feature spaces. In HMRec, domain-shared and domain-specific features are extracted to enable the knowledge transfer between multiple heterogeneous source and target domains. To ensure positive transfer, the domain-shared subspaces from multiple domains are maximally matched by a multiclass domain discriminator in an adversarial learning process. The recommendation in the target domain is completed by a matrix factorization module with aligned latent features from both the user and the item side. Extensive experiments on four cross-domain recommendation tasks with real-world datasets demonstrate that HMRec can effectively transfer knowledge from multiple heterogeneous domains collaboratively to increase the rating prediction accuracy in the target domain and significantly outperforms six state-of-the-art non-transfer or cross-domain baselines.**

***Index Terms*— Adversarial learning, cross-domain recommendation recommender systems, knowledge transfer, recommender systems.**

I. Introduction

RECOMMENDER systems are developed to handle the information overload problem and aim at automatically identifying user preferences with historical data and providing users with personalized services [1]. In addition to the content-based [2] and knowledge-based [3] recommendation techniques, collaborative filtering (CF) plays a vital role in the development of recommender systems, which assumes that users who have similar tastes in the past may also share similar preferences in the future [4]. CF has been extensively investigated in academia and widely applied in the industry from basic memory-based CF to diverse model-based CF, including shallow methods and deep methods [5]. Of the various shallow methods in CF, matrix factorization (MF) projects users and items into a low-dimensional latent feature space and predicts user preference with low-rank approximation [6]. Deep CF methods take advantage of the ability of neural networks to model nonlinear relationships between users and items and have therefore achieved competitive performance [7]–[9].

Though they have evolved rapidly, CF methods suffer from the data sparsity problem, which impairs the performance of recommender systems in real-world applications. A solution to this problem is to transfer knowledge from a source domain with relatively rich data, which is known as cross-domain recommender systems. There are two types of cross-domain recommender systems depending on how common knowledge is shared across domains: one is to connect two domains with auxiliary information, such as user reviews [10], social network [11], and semantic knowledge [12]. However, for domains with insufficient data, the auxiliary information is often not available either. The other type is to transfer knowledge through extracted latent features via user/item overlapping or similarity [13], [14]. Some methods require a full overlap of users or items in both source and target domains, so the overlap part can act as a bridge to transfer knowledge [15]–[17]. However, a full overlap is a strong assumption that is difficult to achieve in real-world applications. The correspondence of users or items in different domains is usually unavailable due to user privacy concerns. Some cross-domain recommendation methods can deal with the non-overlapping scenario by extracting common knowledge based on collective user behavior in a group-level manner [18] or use domain adaptation methods like adversarial learning [19]. These methods can assist the recommendation task in the target domain, but their performance is impaired when data in the source domain is sparse (see results in Tables III and IV).

Existing cross-domain recommendation methods mostly take advantage of one single-source domain with relatively dense data but they are not able to extract useful knowledge when data is sparse in the source domain. In practice, data is usually scattered in heterogeneous forms in similar but

Manuscript received September 21, 2020; revised February 21, 2021; accepted February 16, 2022. This work was supported by the Australian Research Council (ARC) through the Australian Laureate Fellowship under Grant FL190100149. *(Wenhui Liao and Qian Zhang contributed equally to this work.) (Corresponding author: Jie Lu.)*

Wenhui Liao is with the Australian Artificial Intelligence Institute, Faulty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia, and also with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 518071, China (e-mail: lwh18@mails.tsinghua.edu.cn).

Qian Zhang, Guangquan Zhang, and Jie Lu are with the Australian Artificial Intelligence Institute, Faulty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: qiang.zhang-1@uts.edu.au; guangquan.zhang@uts.edu.au; jie.lu@uts.edu.au).

Bo Yuan is with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 518071, China (e-mail: yuanb@sz.tsinghua.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3154345.

Digital Object Identifier 10.1109/TNNLS.2022.3154345

different domains, which are all sparse and complement each other. For example, users who rated "Titanic" with five-star have also liked a book "Jane Eyre" and listened to "My heart will go on." From only one domain (movie/book/music domain), it is hard to know whether users prefer romantic movies/books/music, but together we can come to that conclusion. To alleviate data sparsity and improve the recommendation performance in a sparse target domain, we need to exploit data from multiple domains that can benefit the target domain. Though some cross-domain recommendation methods can be extended to the multidomain scenario with some adjustments [18], [20], they lack in-depth investigation and therefore cannot achieve high accuracy. Also, most methods focus on only homogeneous data, either explicit numerical data [21] or the implicit binary data [22]. When data in the target and source domains are heterogeneous, knowledge transfer becomes more difficult due to the larger discrepancy of data distributions [23].

Hence, to achieve knowledge transfer and cross-domain recommendation with multiple source domains, there are two challenging issues that have not yet been solved in the existing literature: 1) data in a single-source domain is often sparse as well and not sufficient to support the recommendation in the target domain, while data from multiple source domains are often available. How to extract knowledge from multiple source domains that work as a mutual supplement to each other to improve the performance of the recommendation in the target domain remains a challenging and unsolved problem and 2) data in different domains is usually heterogeneous. Data in each domain can be binary interactions or numerical ratings, thus we could have homogeneous source domains but different from the target domain or even heterogeneous source domains. The latent feature extracted from those data lacks direct correlation and lies in different feature spaces with large discrepancies. Therefore, constructing an appropriate shared feature space for each domain is extremely challenging and matters with model performance.

In this article, we propose a novel deep heterogeneous multidomain recommendation system, which aims to improve the rating prediction accuracy of the recommendation in the target domain by transferring shared knowledge from multiple heterogeneous source domains. In our proposed method, two kinds of encoders are designed for domain-specific and domain-shared feature extraction in each domain, where an orthogonality constraint is applied to separate their latent feature subspaces. Through an adversarial training process of the domain-shared feature with the proposed multiclass domain discriminator, the domain-invariant features are extracted and the shared feature spaces are maximally matched. To ensure the effectiveness of the features extracted from heterogeneous source domains, the latent features learned from explicit numerical ratings or implicit binary interactions are, respectively, reconstructed by a decoder with mean squared loss or a classifier with cross-entropy loss. The recommendation task in the target domain is completed by a matrix factorization module with latent features from both the user and the item sides, which combine both the maintained domain-specific features and the domain-shared features learned from multiple sources. These ideas are sharpened into the proposed deep heterogeneous multidomain recommendation method (HMRec). The main contributions of this article are as follows.

1) We propose a novel multidomain recommendation method with an adversarial network, which addresses the deficiency of knowledge from only one source domain. Through the multiclass domain discriminator and the adversarial learning mechanism, the latent feature spaces are maximally matched and then the domain-invariant features are extracted and shared across domains.
2) We achieve knowledge transfer from heterogeneous source domains to improve recommendation in the target domain, especially learning from implicit binary data to assist explicit rating prediction. By modeling heterogeneous data and matching their inconsistent latent feature spaces, we construct a shared latent space collectively and share data-independent knowledge, which ensures that the knowledge from heterogeneous sources is comprehensive and effective.
3) The proposed HMRec is a flexible network that is able to deal with heterogeneous data from multiple domains. Extensive experiments on four tasks with two sparsity levels show that our proposed method outperforms state-of-the-art baselines on the rating prediction tasks for the cross-domain recommendation. The ablation study further confirms the effectiveness of each part of HMRec.

The remainder of this article is organized as follows. Section II introduces some key related work. Section III gives the formal problem definition and preliminaries. In Section IV, we present the details of our proposed method HMRec. Experiments on four real-world datasets are presented in Section V, followed by the ablation study as well as the parameters and complexity analysis. The conclusion and future studies are given in Section VI.

II. Related Work

This section will review three related areas of research.

A. Cross-Domain Recommender Systems

Recommender systems have been developed rapidly in the era of information explosion and have successfully applied in many areas [24]. However, data sparsity is still one of the most challenging issues in recommender systems, which hinders its development, especially for the newly launched ones. Cross-domain recommender systems aim to alleviate the data sparsity problem and use the information from source domains that have relatively sufficient data, which can be seen as the intersection of transfer learning and recommendation systems. Nevertheless, the mainstream of transfer learning is focused on images, and the techniques in transfer learning are difficult to be directly applied to various recommender systems.

Some cross-domain recommender systems are developed with MF techniques, and most of them focus on explicit data. Collective matrix factorization (CMF) [20] jointly factorizes source and target matrices by sharing the user latent features. The rating matrix generated model (RMGM) [18] is a probabilistic model using the soft membership in groups instead of

using the hard membership like an earlier version [25]. Some methods try to achieve heterogeneous recommendation [26], [27] by using a tri-factorization way to capture the low-rank sharing rating patterns, which requires that the number of users and the number of items are exactly the same in both the source and the target domains. The cross-domain recommender system with consistent information transfer (CIT) [21] tries to achieve domain adaptation by mapping generated user groups in both domains and adjusting them to maintain consistency during the knowledge transfer process.

As for deep learning methods, more works focus on the implicit data. Deep cross networks are employed to learn and transfer shared rating patterns among domains [28], which is limited by the same user assumption. To incorporate the auxiliary information, [29] uses a Bayesian neural network to realize a general cross-domain framework, making use of the low-dimensional representation from both the one-hot encoding and auxiliary information of users/items. Domain Adaptation for cross-domain Recommendation via transferring rating patterns (DARec) uses autoencoders (AEs) to generate embeddings to represent each user and interleave them to feed into a domain adaptation module for the shared user rating patterns [30]. However, none of these methods can take advantage of multiple source domains, let alone ensuring positive transferring from heterogeneous and sparse source domains.

B. Adversarial Learning

Adversarial learning is well known for generative modeling in generative adversarial networks (GANs), which can be seen as a two-player game between the generator and the discriminator [31]. Recently, several difficulties of GANs, such as the ease of training or avoiding mode collapse, have been addressed [32]. Meanwhile, adversarial learning has been applied to reduce the discrepancy of data distributions across domains [33], [34], which can be embedded into deep networks to learn the transferable features [35]. The method proposed by [36] extends GANs to multiple discriminators, which can significantly enhance the distribution matching process. Another multiadversarial domain adaptation approach also shows competitive performance, which enables fine-grained alignment of different data distributions by using multiple discriminators to captures multimode structures [35]. A novel algorithm proposed by [19] transfers the latent features from a source domain to a target domain in an adversarial way to alleviate the data sparsity and imbalance issues. Adversarial learning techniques can help us to bridge and align different domains by learning the domain-invariant representations, which can be treated as domain-shared features for the purpose of knowledge transfer in cross-domain recommender systems.

C. Cross-Domain Knowledge Transfer

Cross-domain knowledge transfer aims to improve a model's performance on the target domain by leveraging knowledge from source domains [37], [38]. Early transfer learning studies often intuitively use source domains data for model pretraining. Due to the distribution shift in both the conditional probability and marginal distribution across domains, domain adaptation methods are widely studied recently as a typical type of transfer learning and have shown to be highly competent in various computer vision or natural language processing tasks [39]. As probability distributions may vary from source to source, one solution is proposed by [40] to make different views from various sources complementary to each other through a co-training method while minimizing the distribution divergence.

The source domains maybe heterogeneous to the target, which increases the difficulty of domain adaptation and knowledge transfer. A novel heterogeneous domain adaptation method [41] presents progressive alignment by sharing latent coding and narrowing the distribution gaps to deal with various features in arbitrary dimensions. Cross-domain landmark selection [42] is able to learn from cross-domain data with different types of features by identifying representative cross-domain data, whose adaptation capabilities can be determined accordingly. A newly launched method proposed in [43] ensures effective knowledge transfer by applying a Grassmann—linear geodesic flow kernel, which is designed for heterogeneous unsupervised domain adaptation. Multidomain knowledge transfer techniques are needed in cross-domain recommendation methods, especially when the feature space in the target domain and the source domains are quite different. However, in recommender systems without explicit feature space, the alignment is usually achieved by the overlap between users or items. In the non-overlap scenario, knowledge transfer is achieved by the alignment of the latent feature spaces, which is quite different from the explicit feature spaces of images in transfer learning. Therefore, straightforward applications of transfer learning methods usually fail to work.

III. Preliminaries

We first introduce notations used throughout this article and formally define the problem. Then we introduce a basic AE for collaborative filtering in a single domain as the preliminaries.

A. Notations and Problem Definition

For notations in this article, we use bold letters to denote vectors, capital bold letters to denote matrices. The relationship between users and items in each domain is represented by the user–item pair (m, n) and the corresponding explicit rating $r(m, n)$ or implicit record $y(m, n)$. Any explicit rating r in the corresponding rating matrix $\boldsymbol{R}$ is subject to $r \in \{1, 2, 3, 4, 5, ?\}$ ("?" for unobserved rating). An indicator matrix $\boldsymbol{W}$ can be denoted to represent whether there is a rating between the user and the item or not. As for the domain with implicit data, we use $\boldsymbol{Y}^+$ denotes a set of the observed positive interaction, and $\boldsymbol{Y}^-$ denotes the set of negative instances, which is sampled from the unobserved user–item interactions like [7]. Any implicit rating in the corresponding interaction matrix $\boldsymbol{Y}$ is subject to $y(m', n') \in \{0, 1\}$ ("0" refers to the sampled negative instance, "1" refers to the observed interactions). Then $\boldsymbol{Y}^+ \cup \boldsymbol{Y}^-$ means all training interactions for the implicit source domain.

From the matrix $\boldsymbol{R}$, each user m is represented by a rating vector as $\boldsymbol{r}_u(m, *)$, which represents the user m's ratings across

TABLE I
FREQUENTLY USED NOTATIONS

Notation	Description
$t = \{U^t, I^t, \boldsymbol{R}^t\}$	The target domain
$\{sj\}_{j=1}^d = \{U^{sj}, I^{sj}, \boldsymbol{R}^{sj}\}_{j=1}^d$	The multiple source domains
$\mathbf{S}_r$	The union of source domains with explicit data
$\mathbf{S}_b$	The union of source domains with implicit data
$\boldsymbol{R}$	Explicit numerical rating matrix
$\boldsymbol{Y}$	Implicit binary rating matrix
$\boldsymbol{r}_u/\boldsymbol{r}_i$	Explicit user/item vector
$\boldsymbol{y}_u/\boldsymbol{y}_i$	implicit user/item vector
$\boldsymbol{p}_u/\boldsymbol{p}_i$	Domain-specific user/item feature vector
$\boldsymbol{c}_u/\boldsymbol{c}_i$	Domain-shared user/item feature vector
$\boldsymbol{l}^{s1}, \ldots, \boldsymbol{l}^{sd}, \boldsymbol{l}^t$	Domain label
$\boldsymbol{f}$	Combined latent feature vector
G_e	Feature extractor
G_d	Multi-class domain discriminator
G_r	Reconstructor for heterogeneous source domains
G_p	Rating predictor for the target domain
L_o	Loss caused by the orthogonality constraint
L_d	Loss caused by the multi-domain discriminator
L_r	Loss caused by the reconstructor
L_p	Loss caused by the rating predictor
$g(*)$	ReLu function
$\sigma(*)$	Sigmoid function
$\Phi(*)$	Softmax function

all items (i.e., the mth row of the rating matrix), while the rating vector of each item n (i.e., the nth column of the rating matrix) is denoted as $\boldsymbol{r}_i(*, n)$. Similarly, in the implicit matrix, the user interaction vector for user m' is denoted as $\boldsymbol{y}_u(m', *)$ and the item interaction vector for item i can be denoted as $\boldsymbol{y}_i(*, n')$. We can efficiently leverage the observed information in user or item vectors and produces a vector space with a meaningful sub-structure.

Given a target domain $t = \{U^t, I^t, \boldsymbol{R}^t\}$ with explicit feedback and multiple source domains $sj = \{U^{sj}, I^{sj}, \boldsymbol{R}^{sj}\}$, $j = 1, 2, \ldots, d$ with either explicit or implicit feedback, each domain contains a user set, an item set, and a user–item explicit rating matrix $\boldsymbol{R}$ or implicit interaction matrix $\boldsymbol{Y}$. We use $\mathbf{S_r}$ to denote the union of source domains with explicit rating data, while $\mathbf{S_b}$ to denote the union of source domains with implicit binary data. The most frequently used notations are summarized in Table I.

In this article, we aim to develop a multidomain recommendation method to assist the recommendation task of predicting the ratings in a target domain with explicit ratings using knowledge from all the rating matrices in the other domains served as the source domains. To make full use of the data and ensure positive transfer, both domain-specific knowledge and domain-shared knowledge should be taken into consideration, where the domain-specific knowledge means the unique property of a certain domain that should be maintained and the domain-shared knowledge is the domain-invariant knowledge that can be meaningful for all the domains.

In our problem setting, no observable correspondence exists on the users/items across the domains and users/items are treated as completely different. (In fact, there may exist some correspondences, but the alignment is unavailable due to user privacy.) The problem is formally defined as follows.

The cross-domain recommendation task is to predict the missing values in $\boldsymbol{R}^t$ with the assistance of the source domains. In our problem setting, we assume that $\forall j, U^{sj} \cap U^t = \varnothing, I^{sj} \cap I^t = \varnothing$.

B. Preliminary of Autoencoder

Here, we present a basic method, whose structures will be used for feature extraction or rating reconstruction in our proposed method. The basic method AE for collaborative filtering in a single domain includes an encoder part as well as a decoder part. We take a sparse vector $\boldsymbol{x}$ as the input and leverage an encoder to map it into a low-dimensional space to get a dense vector $\boldsymbol{h}$

$$\boldsymbol{h} = g_1(\boldsymbol{w}_1\boldsymbol{x} + \boldsymbol{b}_1). \tag{1}$$

Then we can predict unknown ratings by using a decoder to recover the rating vector

$$\hat{\boldsymbol{x}} = g_2(\boldsymbol{w}_2\boldsymbol{h} + \boldsymbol{b}_2). \tag{2}$$

Here, $g_1(*)$ and $g_2(*)$ are activation functions (we use ReLu function in this article), and $\boldsymbol{w}_1, \boldsymbol{w}_2$ are the weights and $\boldsymbol{b}_1, \boldsymbol{b}_2$ are the bias for the AE structure. There can be multiple hidden layers included in the AE structure if needed. The AE is applied to deal with the recommendation task with explicit data $\boldsymbol{R}$ [44]. We can take the rating vector of each user or each item as the input and leverage an encoder to map it into a low-dimensional feature space to get the user or latent representation. Then we use a decoder to reconstruct the latent features vectors to a completed rating matrix $\hat{\boldsymbol{R}}$. The parameters can be trained with back-propagation by minimizing the mean squared error of the observed ratings

$$\min \|\boldsymbol{W} \odot (\hat{\boldsymbol{R}} - \boldsymbol{R})\|^2 \tag{3}$$

where $\odot$ denotes the element-wise product of matrices.

C. Preliminary of Adversarial Learning

Here, we present a basic network structure that can be used for an adversarial learning procedure. The adversarial training process can be achieved via a gradient reversal layer [33], [45]. It has different behaviors during the forward and backward propagation processes, which acts as the identity function for forward propagation, but reverses the gradient direction for backward propagation. Learning with a gradient reversal layer is adversarial in that the parameters can be optimized to increase a classifier's ability to discriminate between representations from the source or target domains, while the reversal of the gradient results in the model parameters learning representations from which domain classification accuracy is reduced

$$\begin{aligned} G(x) &= x \quad \text{(forward propagation)} \\ \partial G(x)/\partial x &= -\lambda \boldsymbol{I} \quad \text{(backward propagation)} \end{aligned} \tag{4}$$

where $\boldsymbol{I}$ is an identity matrix and λ is a hyperparameter that can be either a constant or a parameter that changes with data iteration. Unlike attribute features in image processing in transfer learning studies [33], [45], the latent features are learned from preference data in this article. Therefore, we cannot use FFT, normalization, or other transformation way to mitigate the chance of adversarial error, in which case constructing an appropriate shared feature space for each domain is extremely challenging, and direct transformation is incompetent for this task. In this article, we use adversarial learning in an end-to-end network designed to minimize its discrepancy, and the gradient reverse layer can be part of the adversarial learning component.

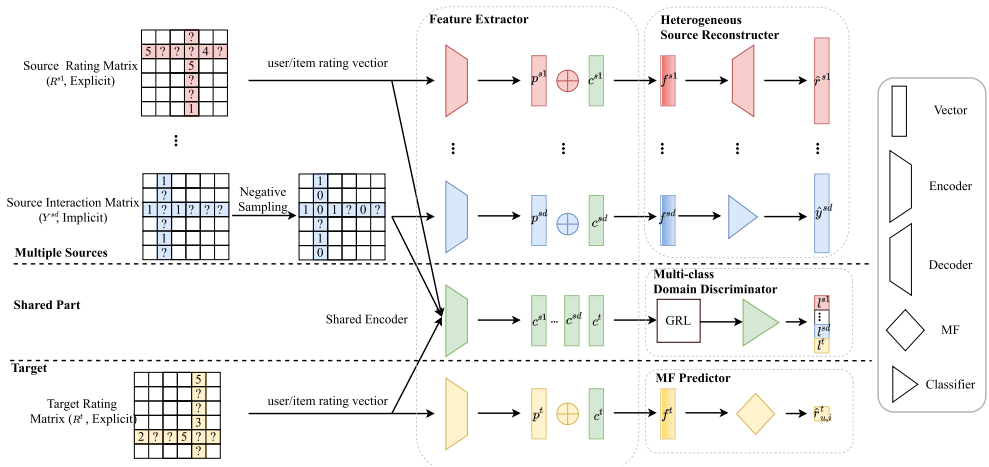


Fig. 1. Proposed architecture of the HMRec method. The target domain is in yellow, the domain-shared part is in green, and a source domain with explicit data is in red, while another source domain with implicit data is in blue. It is an end-to-end feed-forward neural network consisting of four components: 1) the feature extractor part, including multiple domain-specific encoders for each domain and a domain-shared encoder; 2) the domain discriminator part; 3) the rating reconstruction part for the source domains; and 4) the MF predictor part for the target domain.

IV. Heterogeneous Multidomain Recommendation

HMRec is a deep adversarial network for cross-domain recommendation capable of learning from multiple source domains to assist the rating prediction task in the target domain, as shown in Fig. 1. It is an end-to-end feed-forward neural network consisting of four different components, where the information flows from partially observed user–item interaction or rating vectors (input) to predicted ratings (output). The four kinds of components include the feature extractor part, the domain discriminator part (the gradient reverse layer along with the multidomain classifier), the rating reconstruction part for the source domains, and the MF predictor part for the target domain.

A. Feature Extractor G_e

The feature extractor G_e with parameter set θ_e contains several encoders, including domain-specific encoders for domain-specific information preserving as well as domain-shared encoders for cross-domain knowledge transfer, which can deal with heterogeneous input from multiple domains of both user and item sides. We have got one specific encoder for each domain (i.e., d specific encoders for d source domains) and one shared encoder for all domains in our proposed method, and the encoders are used to extract the low-dimensional latent factors for each user or item. Since the latent feature spaces among the source domains and the target domain are quite different, two kinds of encoders in each domain are needed: one is to extract domain-shared features, and the other is to extract domain-specific features. To achieve this goal, we cut off certain links of the hidden layers and divide them into several parts, through which the domain-shared and domain-specific parts are captured and preserved. We take the user side as an example to explain the detailed structure. The item-side part is similar and omitted.

The domain-specific part is only linked and trained by data from a single domain. The target domain-specific feature vector $\boldsymbol{p}_u^t(m)$ of user m is encoded by his/her ratings $\boldsymbol{r}_u^t(m,*)$ in the target domain. The source domain-specific feature vectors $\boldsymbol{p}_u^{s1},\ldots,\boldsymbol{p}_u^{sd}$ in multiple source domains are encoded similarly. The domain-shared part is a common part and trained by data from all the target and source domains, where the domain-shared feature vectors $\boldsymbol{c}_u^t$, $\boldsymbol{c}_u^{s1},\ldots,\boldsymbol{c}_u^{sd}$ are extracted. To ensure that the domain-specific feature subspace is different from the domain-shared feature subspace, we adopt a loss function with orthogonality constraint [33] to maximize the difference between these subspaces

$$L_{o\_u} = \left\|\left(\boldsymbol{P}_u^t\right)^T \boldsymbol{C}_u^t\right\|_F^2 + \sum_{j=1}^{d}\left\|\left(\boldsymbol{P}_u^{sj}\right)^T \boldsymbol{C}_u^{sj}\right\|_F^2. \tag{5}$$

where $(*)^T$ indicates the transpose of a matrix and $\|*\|_F^2$ is the squared Frobenius norm. The rows of matrices $\boldsymbol{P}_u^t$, $\boldsymbol{P}_u^{s1},\ldots,\boldsymbol{P}_u^{sd}$, are user domain-specific feature vectors $\boldsymbol{p}_u^t$, $\boldsymbol{p}_u^{s1},\ldots,\boldsymbol{p}_u^{sd}$, while rows of matrices $\boldsymbol{C}_u^t$, $\boldsymbol{C}_u^{sd},\ldots,\boldsymbol{C}_u^{sd}$, are domain-shared features $\boldsymbol{c}_u^t$, $\boldsymbol{c}_u^{sd},\ldots,\boldsymbol{c}_u^{sd}$. We can extract both domain-specific and -shared features for items with a similar constraint

$$L_{o\_i} = \left\|\left(\boldsymbol{P}_i^t\right)^T \boldsymbol{C}_i^t\right\|_F^2 + \sum_{j=1}^{d}\left\|\left(\boldsymbol{P}_i^{sj}\right)^T \boldsymbol{C}_i^{sj}\right\|_F^2. \tag{6}$$

Therefore, the total orthogonality constraint loss for the domain-specific and -shared latent features is

$$L_o = L_{o\_u} + L_{o\_i}. \tag{7}$$

B. Multiclass Domain Discriminator G_d

The domain discriminator G_d with parameter set θ_d contains two multidomain classifiers and gradient reverse layers for both the user and the item sides, which aims to match the latent feature spaces and enhance positive knowledge transfer through adversarial training. For each side, the domain-shared encoder is trained to produce a feature vector that a classifier cannot reliably decide which domain the vector is from, while the domain classifier is trained to make accurate discrimination. The gradient reversal layer introduced as (4) is placed between the domain-shared encoder and the domain classifier.

The user-side domain classifier is trained for mapping the shared feature vectors $\boldsymbol{c}_u^t$ and $\boldsymbol{c}_u^{s1},\ldots,\boldsymbol{c}_u^{sd}$ to the d-dimensional domain label score vectors $\boldsymbol{z}_u$. Then we use the softmax

function $\boldsymbol{\Phi}$ to turn the scores into probabilities and calculate the predicted domain label $\hat{\boldsymbol{l}}_u$

$$\hat{\boldsymbol{l}}_u = \boldsymbol{\Phi}(\boldsymbol{z}_u), \boldsymbol{\Phi}(z(i)) = \frac{\exp(z(i))}{\sum_{j=1}^{d} \exp(z(j))}. \tag{8}$$

We use our generated probabilities vector $\hat{\boldsymbol{l}}_u$ to match the d-dimensional true probabilities (the ground-truth domain label) $\boldsymbol{l} \in \mathbb{R}^{1\times d}$, where the dimension 0 stands for the target domain and $1, 2, \ldots, d$ stands for the source domains. The domain adversarial similarity loss, that is, the cross-entropy of the domain labels, is applied as constraints on the shared feature vectors [34]. After the adversarial training process, the user latent feature space of the source domain is matched to the target domain. As another important basis for recommendation tasks, the item latent feature spaces should be matched as well, which is achieved in the same way. Through dual adversarial training from both user and item sides, we maximally match the latent feature spaces to enhance positive knowledge transfer. The loss for the dual-side domain discrimination task can be written as

$$L_d = -\sum_{u=1}^{|U|}\sum_{j=1}^{d} \boldsymbol{l}(j)\ \log \hat{\boldsymbol{l}}_u(j) - \sum_{i=1}^{|I|}\sum_{j=1}^{d} \boldsymbol{l}(j)\ \log \hat{\boldsymbol{l}}_i(j) \tag{9}$$

where $U = U^t \cup U^{s1} \cup \cdots \cup U^{sd}$, $I = I^t \cup I^{s1} \cup \cdots \cup I^{sd}$, $|*|$ means the number of the user or item set, and $\boldsymbol{l}(j)$ means the jth dimension of $\boldsymbol{l}$.

C. Heterogeneous Source Reconstructor G_r

The reconstructor G_r for the source domains with parameter set θ_r contains two kinds of decoders for explicit rating vectors and implicit interaction vectors. G_r aims to reconstruct the rating or interaction vectors, which is used to ensure that the feature vectors from the source domain are still reasonable and avoid trivial solutions. For the source domain, the user-side and item-side feature vectors are reconstructed separately. For the user side, we combine the domain-specific feature vector $\{\boldsymbol{p}_u^{sj}\}_{j=1}^{d}$ and the domain-shared feature vector $\boldsymbol{c}_u^{s1}, \ldots, \boldsymbol{c}_u^{sd}$ to get the user feature vector $\{\boldsymbol{f}_u^{sj}\}_{j=1}^{d}$ before putting them into the reconstructor. The item feature vectors are obtained similarly. When the source domain data is the explicit numerical ratings, we can reconstruct them by a decoder like (2) and (3) mentioned in the preliminaries. For the source domain with implicit binary interactions matrix $\boldsymbol{Y}$, predicted elements in the reconstructed matrix $\hat{\boldsymbol{Y}}$ are required to be constrained in the range of [0, 1], which can be realized by a classifier using a sigmoid function $\sigma(*)$

$$\hat{y}(m', n') = \frac{1}{2}\boldsymbol{I}(n') \cdot \sigma(\hat{\boldsymbol{y}}_u(m', *)) + \frac{1}{2}[\boldsymbol{I}(m')]^T \cdot \sigma(\hat{\boldsymbol{y}}_i(*, n')) \tag{10}$$

where $\boldsymbol{I}$ is a one-hot encoding indicator to represent the position of the training instance. Parameters can be trained with back-propagation by minimizing the cross entropy loss

$$L_{r\_i}(Y) = -\sum_{(m',n')\in \boldsymbol{Y}^*} [\hat{y}(m', n')\ \log y(m', n') + (1 - \hat{y}(m', n')) \log(1 - y(m', n'))] \tag{11}$$

where $\boldsymbol{Y}^+$ is the observed instances of $\boldsymbol{Y}$ and $\boldsymbol{Y}^-$ is the negative instances achieved through negative sampling [7]. Together, the rating reconstructor for both sides of the heterogeneous source domain are trained by optimizing the reconstruction loss

$$L_{r\_e}(R) = \left\| \boldsymbol{W}^{sj} \odot \left(\hat{\boldsymbol{R}}_u^{sj} - \boldsymbol{R}^{sj}\right)\right\|^2 + \left\| \boldsymbol{W}^{sj} \odot \left(\hat{\boldsymbol{R}}_i^{sj} - \boldsymbol{R}^{sj}\right)\right\|^2 \tag{12}$$

where $\boldsymbol{W}^s$ is the indicator matrix of the source domain, and $\hat{\boldsymbol{R}}_u^s$ is the reconstructed rating matrix of the source domain from the user side and $\hat{\boldsymbol{R}}_i^s$ is the one reconstructed from the item side.

Together, the rating reconstructor for both sides of the heterogeneous source domain is trained by optimizing the reconstruction loss

$$L_r = -\sum_{sj\subseteq \mathbf{S_b}} L_{r\_im}(Y^{sj}) + \sum_{sj\subseteq \mathbf{S_r}} L_{r\_ex}(R^{sj}). \tag{13}$$

D. MF Predictor G_p

The MF predictor for the target domain G_p with parameter set θ_p contains a deep MF structure. The user feature vectors and item feature vectors that combine both the domain-specific and the domain-shared features are fed into this module and output the predicted target rating vector. Dimension reduction is further achieved through a linear mapping, and user vector $\boldsymbol{u}^t(m, *)$ for user m as well as item vector $\boldsymbol{i}^t(*, n)$ for item n are obtained. It projects both users and items onto the same latent space so that they are comparable. Then through their inner products, we can predict the unobserved parts. The rating prediction is calculated by

$$\begin{aligned} \boldsymbol{f}_u^t &= \boldsymbol{p}_u^t \oplus \boldsymbol{c}_u^t, \quad \boldsymbol{f}_i^t = \boldsymbol{p}_i^t \oplus \boldsymbol{c}_i^t, \boldsymbol{u}^t = g\left(\boldsymbol{w}_u \boldsymbol{f}_u^t + \boldsymbol{b}_u\right) \\ \boldsymbol{i}^t &= g\left(\boldsymbol{w}_i \boldsymbol{f}_i^t + \boldsymbol{b}_i\right), \quad \hat{r}^t(m, n) = \boldsymbol{u}^t(m, *) \cdot \boldsymbol{i}^t(*, n). \end{aligned} \tag{14}$$

The prediction loss for the target domain can be calculated as

$$L_p = \|\boldsymbol{W}^t \odot (\hat{\boldsymbol{R}}^t - \boldsymbol{R}^t)\|^2 \tag{15}$$

where $\boldsymbol{W}^t$ is the indicator matrix of the target domain and $\hat{\boldsymbol{R}}^t$ is the predicted rating matrix for the recommendation task in the target domain.

The HMRec algorithm is summarized in Algorithm 1. Especially, lines 5–7 aim to extract the latent features through the encoders and optimize them through the adversarial learning process as line 6. Lines 8–13 aim to reconstruct the hidden features for each domain. Line 9 fulfills the rating matrix in the target domain to make rating predictions. Lines 11 and 13 are set to ensure that the representations from source domains are still reasonable and avoid trivial solutions, namely for explicit and implicit data. Lines 15–19 aim to calculate the loss of each component and the whole network can be trained through back-propagation in line 20. The training procedure aims to collaboratively find both domain-specific features for each domain and domain-shared features of multiple heterogeneous domains through deep adversarial learning. It involves optimizing the parameter sets θ_e of G_e to maximize the loss of the domain discriminator as well as minimize the loss of subspace orthogonality constraint (L_o), while the parameter sets θ_d are

Algorithm 1 HMRec Algorithm

Input:
Multiple source domains with explicit or implicit data: $\{sj\}_{j=1}^{d} = \{U^{sj}, I^{sj}, \boldsymbol{R}^{sj} \ or \ \boldsymbol{Y}^{sj}\}_{j=1}^{d}$;
The target domain data:
$t = \{U^t, I^t, \boldsymbol{R}^t\}$, the target domain;

Output:
$\hat{\boldsymbol{R}}^t$, the predicted rating matrix in target domain;

Randomly initialization of the network parameters and set the hyper parameters.
while epoch form 1 to N **do**
  for data in each domain:
    Randomly shuffle the training data
    Extract latent feature vectors $\boldsymbol{p}$ and $\boldsymbol{c}$ with G_e
    Discriminate the domain label $\hat{\boldsymbol{l}}$ with G_d
    Combine the domain-specific $\boldsymbol{p}$ and domain-shared $\boldsymbol{c}$ to get the latent feature factors
      $\boldsymbol{f_u} = \boldsymbol{p_u} \oplus \boldsymbol{c_u}$ for users
      $\boldsymbol{f_i} = \boldsymbol{p_i} \oplus \boldsymbol{c_i}$ for items
    if the training data is from t **then**
      Make the rating prediction in $\boldsymbol{R}^t$ by G_p
    else if the training data is from $sj \subseteq \mathbf{S_r}$ **then**
      Reconstrut $\hat{\boldsymbol{R}}_u^{sj}$ and $\hat{\boldsymbol{R}}_i^{sj}$ by G_r
    else if the training data is from $sj \subseteq \mathbf{S_b}$ **then**
      Reconstruct $\hat{\boldsymbol{y}}$ by G_r to avoid trivial solutions of the source domains
    end if
  Calculate the orthogonality constraint loss L_o by Eq. (5)-(7) to separate the feature subspaces
  Calculate the domain discrimination loss L_d by Eq. (4)
  Calculate the reconstruction loss for the source domains L_r by Eq. (13)
  Calculate the prediction loss for the target domain L_p by Eq. (15)
  Calculate the overall loss function by Eq. (16)
Update the network parameters through back propagation
end while
return $\hat{\boldsymbol{R}}^t$

learned by minimizing L_d. In addition, the loss of the rating reconstructor and the MF predictor is also minimized. Overall, the total loss function for the proposed method HMRec is defined as

$$L(\theta_e, \theta_p, \theta_r, \theta_d) = L_p(\theta_p, \theta_e) + \alpha \cdot L_r(\theta_r, \theta_e) - \beta \cdot L_d(\theta_d, \theta_e) + \gamma \cdot L_o(\theta_e) \quad (16)$$

where α, β, and γ are the trade-off parameters used to balance the contributions of the four components. The optimization objective can be expressed as

$$(\hat{\theta}_e, \hat{\theta}_p, \hat{\theta}_r) = \arg\min_{\theta_e, \theta_p, \theta_r} L(\theta_e, \theta_p, \theta_r, \theta_d)$$
$$(\hat{\theta}_d) = \arg\max_{\theta_d} L(\theta_e, \theta_p, \theta_r, \theta_d). \quad (17)$$

With the gradient reverse layers, the network parameters can be optimized by stochastic gradient descent or its variants like the adaptive moment method through backward propagation in an end-to-end manner, as shown in Fig. 2. After training

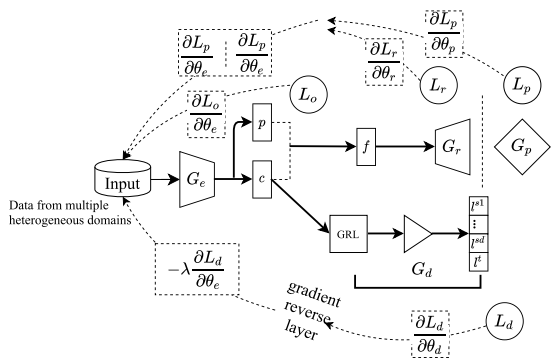


Fig. 2. Feed-propagation and back-propagation process of HMRec, where $\boldsymbol{p}$ is the domain-specific feature, $\boldsymbol{c}$ is the domain-shared feature, and $\boldsymbol{f}$ is the combined latent feature for each domain, and $\boldsymbol{l}^{s1}, \ldots, \boldsymbol{l}^{sd}, \boldsymbol{l}^t$ is the domain label; G_e is the feature extractor, G_d is the multiclass domain discriminator, G_r is the heterogeneous reconstructor for the source domains, and G_p is the rating predictor for the target domain.

convergence, the parameters $\hat{\theta}_e, \hat{\theta}_p, \hat{\theta}_r, \hat{\theta}_d$ will deliver a saddle point of the overall loss function (16).

V. Experiment and Analysis

This section presents the experimental results and the related analysis. The datasets and evaluation metrics are introduced first, followed by the baselines and the experiment settings. Then, we compare the results of the empirical experiments followed by a discussion. We also conduct an ablation study to show the effectiveness of each component in our proposed method. Lastly, a parameter and complexity analysis is presented.

A. Datasets and Evaluation Metrics

1) Datasets: Our experiments are conducted on public datasets, Douban,[1] Amazon,[2] MovieLens,[3] Netflix,[4] and YahooMusic.[5] Each of these datasets is publicly available and has been tested in various situations on both single-domain and cross-domain recommendation tasks, but rarely used in the heterogeneous multidomain recommendation. We preprocessed the datasets to filter out the meaningless records. For Amazon, we choose two groups of categories to conduct the experiment as Tasks 1 and 2. The movie and clothing categories are chosen as the target domains, while the beauty category serve as the source domain with explicit data and the book, CD, and sports categories are processed to be the source domain with implicit data. It should be noted that Amazon datasets contain many users who gave all the items the same rating, which means these user records will make no contribution to the recommendation task [21]. Therefore, we removed these meaningless records and the resulting empty rows and columns. For the Douban dataset, we randomly choose 3000 users and 5000 items for each domain as Task 3. The movie category is chosen as the target domain, while the book category and the music category serve as the source

[1] https://sites.google.com/site/erhengzhong/datasets
[2] http://jmcauley.ucsd.edu/data/amazon/
[3] https://grouplens.org/datasets/movielens/20m/
[4] https://netflixprize.com/index.html
[5] http://webscope.sandbox.yahoo.com/

TABLE II
STATISTICS OF THE DATASETS

Task	Data_name	Domain	Format	User No.	Item No.	Sparsity
Task 1	Amazon_book	Source1	implicit	3000	3000	0.55%
	Amazon_CD	Source2	implicit	3000	3000	0.69%
	Amazon_movie1	Target1	explicit	1990	1854	0.26%
	Amazon_movie2	Target2	explicit	1990	1854	0.23%
Task 2	Amazon_beauty	Source1	explicit	2696	1869	0.46%
	Amazon_sports	Source2	implicit	3362	2284	0.28%
	Amazon_clothing1	Target1	explicit	3000	3000	0.49%
	Amazon_clothing2	Target2	explicit	3000	3000	0.34%
Task 3	Douban_music	Source1	explicit	3000	5000	0.75%
	Douban_book	Source2	implicit	3000	5000	1.18%
	Douban_movie1	Target1	explicit	3000	5000	0.62%
	Douban_movie2	Target2	explicit	3000	5000	0.47%
Task 4	Netflix	Source1	implicit	4000	2000	0.49%
	YahooMusic	Source2	implicit	4000	2000	0.49%
	MovieLens	Target1	explicit	4000	2000	0.50%
	MovieLens	Target2	explicit	4000	2000	0.34%

domains. To fit the problem setting of multiple heterogeneous source domains, the ratings of the second source domain are replaced by implicit binary records. The Netflix and YahooMusic datasets are normalized to the binary data of 0, 1 and serve as the source domain of the target domain MovieLens as Task 4. After that, experiments are conducted in a data sparsity problem setting for the target domains, where ratings are given at least 1 but no more than 4 for the test users and the rest is used for sparsity adjustment and evaluation. Two different sparsity ratios of the target domain for each task are achieved by random sample adjustment similar to [18]. The statistics of these datasets are shown in Table II and the four tasks are listed follows.

1) *Task 1:* Amazon_book, CD → Amazon_movie
2) *Task 2:* Amazon_beauty, sports → Amazon_clothing
3) *Task 3:* Douban_music, book → Douban_movie
4) *Task 4:* Netflix, YahooMusic → MovieLens.

2) Evaluation Metrics: Root mean square error (RMSE) and mean absolute error (MAE) are used as the evaluation metrics.

B. Baselines and Experimental Settings

1) Different Types of Baselines: We compare our proposed method HMRec with two groups of recommendation baselines as shown in Table III. The first group includes three nontransfer recommendation methods.

1) SVD [6] is a famous matrix factorization method, the most popular being the non-transfer shallow method;
2) AutoRec [44] uses AE to reconstruct the unknown ratings from the observed ones;
3) Deep matrix factorization (DMF) [7] is a deep CF method that uses multiple nonlinear layers to extract latent feature vectors of users and items.

The second group has three cross-domain recommendation methods.

1) *DMF++:* We combine two DMFs by simply sharing the user or item latent features as a straight way to transfer knowledge. For the implicit data, the interaction can be predicted by $\hat{y} = \sigma(\boldsymbol{h}_u \cdot \boldsymbol{h}_i)$, where $\sigma(*)$ is a sigmoid function to constrain the output $\hat{y}$ in the range of [0, 1], and the optimization goal will be (11).

TABLE III
BASELINES

Baselines	Shallow Methods	Deep Methods
Single-domain	SVD	AutoRec, DMF
Cross-domain	CMF	DMF++, DARec

2) Collective matrix factorization (CMF) [20] jointly factorizes matrices of individual domains with the constraint that the user/item feature matrix is the same. For the source domain with implicit data, CMF is applied with the logistic link function as discussed in [46]. For the multisource domain scenario, multiple constraints are applied for both the user and item sides.
3) DARec [30] is a state-of-the-art deep domain adaptation model for cross-domain recommendation, which uses AE structures combined with the DANN to transfer knowledge between the extracted user or item rating patterns. For the source domain with implicit data, the loss function of AE is replaced by cross entropy as in [47] and [48] and for the multisource domain scenario, rating patterns extracted from multiple source domains are mixed before being sent to the DANN.

2) Parameter Settings: We use Pytorch to implement our method. All the other methods are from original code (if available) or rewritten to a Pytorch version based on the original article. All parameters are set according to the original article and are fine-tuned. We use a grid search strategy to tune the learning rate (from 0.0001 to 0.1), batch size (from 64 to 1024), and latent factor dimensions (from 10 to 500) to obtain their best performance. For the source domain with implicit data, the negative sampling ratio is set to be similar to the number of positive instances (from 3 to 10).

For the network structure of HMRec, we design two hidden layers for each encoder, three hidden layers for the domain classifier, and two hidden layers for each rating reconstructor. The MF predictor contains three hidden layers. All parameter matrices in the model are initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.01. The loss of classification is ignored in the training of the first three batches to pre-train the other part of the network. For data from the source domain, we use an exponential decay of 0.95 on the learning rate every ten batches. Thus, the influence of the target domain data is gradually increased to fine-tune the recommendation result. We run the experiments ten times and return the mean and standard deviations.

C. Experimental Results

The comparison results of our proposed HMRec method with various baselines on the four tasks in two different sparsity ratios are shown in Tables IV–VI. We compare HMRec with three non-transfer methods to show the effectiveness of knowledge transfer as shown in Table IV. For the cross-domain methods, we conduct the experiment that they transfer knowledge from a single-source domain (marked as with S1 or S2) and from multiple source domains simultaneously (marked as multi) to show the effect of each source domain, whose results

TABLE IV

OVERALL COMPARISON RESULT WITH THE NONTRANSFER BASELINES

	Sparsity	RMSE				MAE			
		SVD	AutoRec	DMF	HMRec	SVD	AutoRec	DMF	HMRec
Task 1	0.26%	1.4430 (±0.007)	1.3619 (±0.017)	1.5082 (±0.042)	**1.2571*** (±0.009)	1.1268 (±0.005)	1.1209 (±0.010)	1.1778 (±0.063)	**1.0184*** (±0.004)
	0.23%	1.4502 (±0.010)	1.3727 (±0.009)	1.5270 (±0.050)	**1.2743*** (±0.004)	1.1129 (±0.010)	1.1352 (±0.019)	1.1868 (±0.057)	**1.0286*** (±0.009)
Task 2	0.49%	1.3969 (±0.007)	1.4397 (±0.015)	1.3939 (±0.037)	**1.2914*** (±0.014)	1.0822 (±0.003)	1.1648 (±0.013)	1.1259 (±0.016)	**1.0489*** (±0.014)
	0.34%	1.3920 (±0.003)	1.4549 (±0.010)	1.4104 (±0.037)	**1.2956*** (±0.008)	1.0812 (±0.014)	1.1726 (±0.008)	1.1392 (±0.050)	**1.0533*** (±0.020)
Task 3	0.62%	1.0841 (±0.018)	0.8801 (±0.023)	1.0794 (±0.041)	**0.8095*** (±0.003)	0.8324 (±0.003)	0.6864 (±0.010)	0.8676 (±0.046)	**0.6394** (±0.005)
	0.47%	1.0904 (±0.004)	0.8655 (±0.025)	1.0773 (±0.054)	**0.8114*** (±0.003)	0.8222 (±0.003)	0.6795 (±0.014)	0.8437 (±0.031)	**0.6405*** (±0.003)
Task 4	0.50%	1.2362 (±0.001)	1.2264 (±0.010)	1.3481 (±0.052)	**1.1846*** (±0.001)	1.0221 (±0.001)	1.0137 (±0.006)	1.1129 (±0.060)	**0.9903*** (±0.001)
	0.34%	1.2629 (±0.002)	1.2566 (±0.021)	1.3759 (±0.021)	**1.1936*** (±0.003)	1.0437 (±0.002)	1.0320 (±0.012)	1.1383 (±0.016)	**0.9945*** (±0.017)

TABLE V

OVERALL COMPARISON RESULT WITH CROSS-DOMAIN BASELINES IN TERMS OF RMSE

	Sparsity	DMF++			CMF			DARec			HMRec		
		with S1	with S2	multi	with S1	with S2	multi	with S1	with S2	multi	with S1	with S2	multi
Task 1	0.26%	1.4914 (±0.032)	1.4554 (±0.029)	1.3866 (±0.014)	1.4186 (±0.012)	1.4042 (±0.013)	1.3702 (±0.014)	1.4289 (±0.033)	1.4396 (±0.022)	1.4186 (±0.017)	1.2696 (±0.006)	1.2676 (±0.005)	**1.2571*** (±0.009)
	0.23%	1.4799 (±0.030)	1.4679 (±0.016)	1.3998 (±0.015)	1.4367 (±0.018)	1.4067 (±0.017)	1.3774 (±0.013)	1.4859 (±0.035)	1.4887 (±0.022)	1.4367 (±0.018)	1.2842 (±0.009)	1.2829 (±0.009)	**1.2743*** (±0.004)
Task 2	0.49%	1.3817 (±0.014)	1.3917 (±0.020)	1.3682 (±0.034)	1.3558 (±0.020)	1.3691 (±0.012)	1.3260 (±0.022)	1.3539 (±0.009)	1.3703 (±0.009)	1.3411 (±0.014)	1.3106 (±0.008)	1.3179 (±0.007)	**1.2914*** (±0.014)
	0.34%	1.3762 (±0.018)	1.3973 (±0.022)	1.3722 (±0.032)	1.3619 (±0.021)	1.3721 (±0.012)	1.3359 (±0.022)	1.3782 (±0.011)	1.3714 (±0.015)	1.3581 (±0.016)	1.3142 (±0.004)	1.3209 (±0.014)	**1.2956*** (±0.008)
Task 3	0.62%	0.9431 (±0.014)	0.9613 (±0.017)	0.9274 (±0.021)	0.9422 (±0.021)	1.0941 (±0.016)	1.0077 (±0.013)	0.9056 (±0.012)	1.0319 (±0.073)	0.8853 (±0.016)	0.8249 (±0.007)	0.8294 (±0.003)	**0.8095*** (±0.003)
	0.47%	0.9514 (±0.018)	0.9603 (±0.019)	0.9151 (±0.019)	0.996 (±0.017)	1.0914 (±0.015)	1.005 (±0.007)	0.9116 (±0.018)	1.0464 (±0.053)	0.8918 (±0.019)	0.8278 (±0.008)	0.8318 (±0.004)	**0.8114*** (±0.003)
Task 4	0.50%	1.4481 (±0.052)	1.4012 (±0.022)	1.2818 (±0.008)	1.2616 (±0.010)	1.2505 (±0.010)	1.2291 (±0.014)	1.3188 (±0.018)	1.3345 (±0.013)	1.3116 (±0.020)	1.2143 (±0.003)	1.2132 (±0.002)	**1.1846*** (±0.001)
	0.34%	1.4375 (±0.041)	1.4031 (±0.031)	1.3572 (±0.013)	1.281 (±0.019)	1.2831 (±0.019)	1.2366 (±0.023)	1.3750 (±0.011)	1.3730 (±0.013)	1.3210 (±0.019)	1.2228 (±0.003)	1.2217 (±0.002)	**1.1936*** (±0.003)

TABLE VI

OVERALL COMPARISON RESULT OF CROSS-DOMAIN BASELINES IN TERMS OF MAE

	Sparsity	DMF++			CMF			DARec			HMRec		
		with S1	with S2	multi	with S1	with S2	multi	with S1	with S2	multi	with S1	with S2	multi
Task 1	0.26%	1.1613 (±0.043)	1.1513 (±0.025)	1.1351 (±0.019)	1.1708 (±0.016)	1.1703 (±0.011)	1.1726 (±0.011)	1.1495 (±0.018)	1.1596 (±0.011)	1.1708 (±0.016)	1.0299 (±0.005)	1.0344 (±0.009)	**1.0184*** (±0.004)
	0.23%	1.1629 (±0.048)	1.1399 (±0.026)	1.1329 (±0.017)	1.1803 (±0.015)	1.1845 (±0.015)	1.1819 (±0.014)	1.1863 (±0.036)	1.2067 (±0.009)	1.1803 (±0.015)	1.0383 (±0.011)	1.0423 (±0.006)	**1.0286*** (±0.009)
Task 2	0.49%	1.0989 (±0.019)	1.1196 (±0.019)	1.0927 (±0.039)	1.1696 (±0.013)	1.1629 (±0.013)	1.1527 (±0.045)	1.1001 (±0.010)	1.1152 (±0.019)	1.1094 (±0.015)	1.0835 (±0.010)	1.1085 (±0.023)	**1.0489*** (±0.014)
	0.34%	1.1179 (±0.014)	1.1379 (±0.024)	1.1017 (±0.034)	1.1722 (±0.020)	1.1822 (±0.018)	1.1648 (±0.056)	1.1106 (±0.009)	1.1129 (±0.014)	1.1187 (±0.018)	1.0831 (±0.026)	1.0956 (±0.030)	**1.0533*** (±0.020)
Task 3	0.62%	0.7621 (±0.021)	0.7921 (±0.020)	0.7316 (±0.020)	0.8212 (±0.025)	0.8204 (±0.014)	0.8187 (±0.015)	0.7127 (±0.009)	0.7451 (±0.026)	0.7131 (±0.017)	0.6473 (±0.003)	0.6512 (±0.003)	**0.6394** (±0.005)
	0.47%	0.7706 (±0.018)	0.7906 (±0.023)	0.7210 (±0.020)	0.8119 (±0.024)	0.8212 (±0.012)	0.8246 (±0.006)	0.7107 (±0.010)	0.7811 (±0.032)	0.7655 (±0.018)	0.6480 (±0.004)	0.6538 (±0.003)	**0.6405*** (±0.003)
Task 4	0.50%	1.1829 (±0.060)	1.1712 (±0.060)	1.1376 (±0.003)	1.0513 (±0.016)	1.0693 (±0.018)	1.0431 (±0.008)	1.0831 (±0.018)	1.0913 (±0.009)	1.0741 (±0.016)	1.012 (±0.004)	1.0124 (±0.003)	**0.9903*** (±0.001)
	0.34%	1.2383 (±0.039)	1.1809 (±0.029)	1.1453 (±0.008)	1.0727 (±0.019)	1.0814 (±0.018)	1.0402 (±0.009)	1.1217 (±0.008)	1.1143 (±0.009)	1.1205 (±0.017)	1.0175 (±0.029)	1.0277 (±0.012)	**0.9945*** (±0.002)

in terms of RMSE and MAE are shown in Tables V and VI. We also conducted a significance analysis using Friedman's test on all pairs of experiments. Statistically significant results are marked with an asterisk (*). Most of the P-values are much smaller than the significance level $\alpha = 0.01$. Our proposed method HMRec performs best under all the test configurations. We make the following observations.

1) Compared With Non-Transfer Methods: Basically, the classic method SVD performs steadily for all tasks and suffers from the data sparsity issue. The deep learning method AutoRec performs the best of the three non-transfer methods for most of the time, which reveals the effectiveness of an ability to learn nonlinearity. However, the deep matrix factorization method DMF, which achieves competitive performance in many scenes [7], fails to perform well when the training data of the target domain is very sparse.

Overall, our proposed method HMRec performs far better than the non-transfer methods as shown in Table IV, especially when the target domain is extremely sparse. Take the comparison result in the four tasks with the more severe sparsity as an example: our proposed method HMRec improves 5.01%–10.95% in terms of RMSE compared with AutoRec, while achieving 8.14%–24.68% improvement compared with DMF. This shows the effectiveness of knowledge transfer

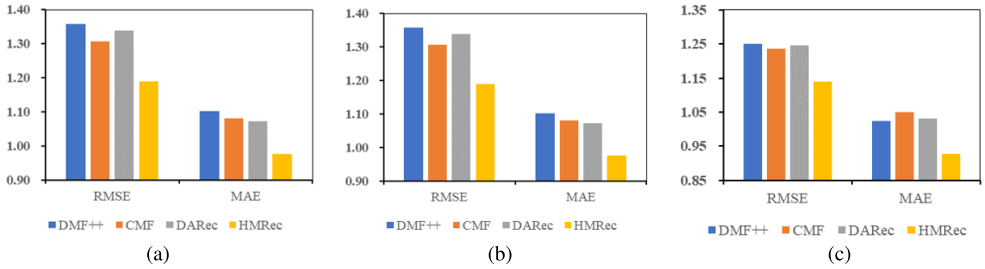


Fig. 3. Average comparison results of the four tasks with the cross-domain methods. (a) Single explicit source domain. (b) Single implicit source domain. (c) Both heterogeneous source domains.

from the source domains. As a result, the proposed method HMRec significantly outperforms all the non-transfer learning recommendation techniques.

2) Compared With Cross-Domain Methods: CMF is a classic cross-domain method and DARec is one of the state-of-the-art methods. Both achieve impressive performance when the source domain is relatively dense and similar to the target domain. However, a negative transfer occurs when the source domain is also quite sparse and even heterogeneous to the target domain. We find that CMF and DARec perform worse than the non-transfer method AutoRec or SVD in Tasks 1 and 4. This is because they only consider domain-shared features while ignoring the domain-specific information of the target domain. In this case, the model shifts more toward the source domain and degrades its prediction performance in the target domain.

Limited by high sparsity, DMF does not perform well. DMF++ achieves better performance than DMF in Tasks 1–3 by introducing richer user information from the source domain. However, in Task 4, a negative transfer occurs and DMF++ results in 8.79% more errors than DMF. Furthermore, the variances of both DMF and DMF++ are relatively larger than the other methods. This is because of the simple replacement of user or item latent features. Our proposed method HMRec achieves from 4.51% to 17.83% improvement on both RMSE and MAE with DMF++ and ensures a steady improvement compared to all the non-transfer learning methods with various sparsity ratios.

To present an overview of the comparison results of the cross-domain methods more intuitively, we show the average comparison results of the four tasks with the cross-domain method in 3. As shown in Fig. 3(a), the cross-domain methods DMF++, CMF, and DARec perform well when the source domain has explicit rating data (e.g. Task 2 with S1 and Task 3 with S2). HMRec demonstrates better performance by achieving 3.20%–16.89% improvement in terms of RMSE and 1.4%–21.18% in terms of MAE. However, as shown in Fig. 3(b), the performance of these baselines is seriously impaired when the source domain is heterogeneous, and our proposed HMRec can achieve more than 20% improvement on several subtasks. For example, in Task 3 with the implicit source domain, DMF++, CMF, and DARec result in 13.72%, 24.19%, and 19.62% more errors than HMRec in terms of RMSE. This reveals that the proposed HMRec can effectively learn from implicit binary data of the source domain to assist explicit rating predictions in the target domain.

3) Comparison With Multiple Sources: The experimental results also reveal the value of transferring knowledge from multiple source domains. Overall, the cross-domain methods achieve better performance when data from multiple source domains are available. However, in Tasks 2 and 3 where the data from S1 is explicit and the data from S2 is implicit, the baselines transferred from both source domains are confused and fail to perform well. Some cross-domain methods may suffer from performance decrease when the number of source domains increases, which is usually caused by severer domain shift and lacking an effective domain adaptation mechanism. For example, in Table VI, the performance of both CMF and DARec in multi is worse than with a single domain (S1 or S2) in Task 1. This phenomenon becomes more apparent when the number of source domains is increased to 4 as shown in Fig. 4, which may be caused by the domain shift. However, the HMRec method proposed in this article aims to improve the recommendation accuracy by taking maximum advantage of the available data and avoiding negative transfer, regardless of the number of source domains. It is very difficult to find the inflection point that no more source domain is needed, or theoretically define which domain or domains are the best. However, thanks to the HMRec network structure we have designed, our method is able to achieve the proposed aim and we have observed no performance degradation during our experiments while the number of source domains increases. In other words, the proposed HMRec can effectively and steadily benefit from multiple heterogeneous source domains by maximally matching their latent feature spaces.

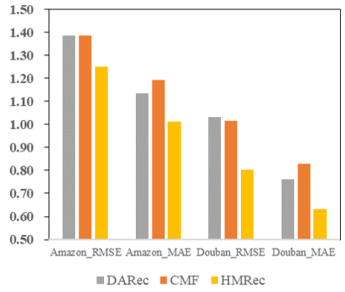
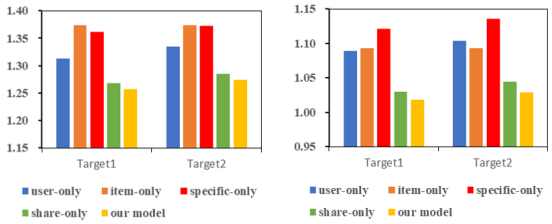


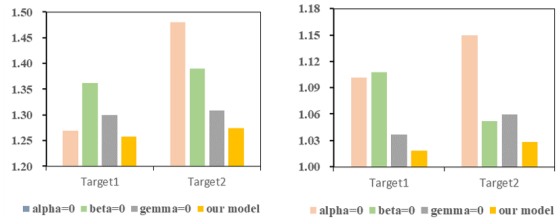
Fig. 4. Results of heterogeneous multidomain recommendation.

As shown in Tables V and VI, HMRec outperforms all the baselines, and its performance remains stable on various datasets under different sparse settings. By transferring knowledge from multiple heterogeneous source domains which are also quite sparse, our proposed method HMRec can effectively learn useful knowledge from the domain-shared features. Furthermore, our method can extract domain-specific features for the target domain, which prevents the occurrence of a negative transfer. When the target domain is extremely sparse or the



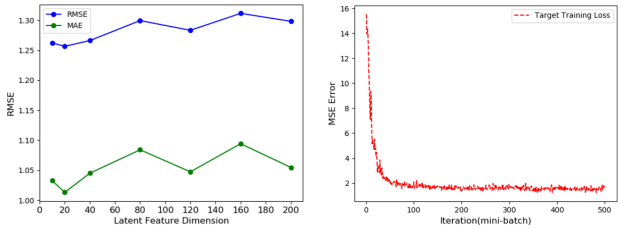
(a) (b)

Fig. 5. Results of the ablation experiments to the network structure. (a) RMSE. (b) MAE.



(a) (b)

Fig. 6. Results of the ablation experiments to the key parameters. (a) RMSE. (b) MAE.



(a) (b)

Fig. 7. Results of parameter analysis and the iteration process. (a) Parameter analysis. (b) Iteration.

source domain is significantly different from it, the advantages of our methods are particularly significant.

D. Ablation Study

Since there are many components in our proposed network, we analyze their impact via an ablation study as shown in Fig. 5. We introduce several component networks as follows.

1) *User_Only:* A network that transfers knowledge only from the user side. We drop the item part of the feature extractor and replace the MF predictor of the target domain with a user rating vector reconstructor similar to the one in the source domains.
2) *Item_Only:* A network that transfers knowledge only from the item side. Therefore, we only take advantage of knowledge transfer from either the user perspective or the item perspective.
3) *Specific_Only:* If we only consider the domain-specific part without the shared part, the network will degenerate into the simple non-transfer method Autorec.
4) *Share_Only:* A network without the domain-specific part. We drop the domain-specific encoders and the loss orthogonality constraint between the shared and specific features is set to 0. Therefore, we only use the shared latent features from the dual adversarial learning to make the recommendation.

From the experimental results on Task 1 with the target domain in two different sparsity ratios shown in Fig. 5, we can observe that our proposed HMRec consistently outperforms the share_only, the specific_only, the user_only, and the item_only one in items of both RMSE and MAE. Specifically, HMRec achieves 7.17%–9.39% improvements compared with the specific_only one, which takes advantage of domain-specific features only and learns nothing from the source domains. This reveals the necessity of our knowledge transfer mechanism. Also, the comparison between HMRec and the share_only one shows the significance of the domain-specific part, where HMRec improves 0.48%–1.08%. Furthermore, for the first target set, HMRec improves 4.26% and 6.53% compared to the user_only network in terms of RMSE and MAE and improves by 8.49% and 6.85% compared to the item_only one. Similarly, HMRec can achieve 4.57%–7.25% improvement compared to the ones on the sparser target set, which reveals the effectiveness of adversarial learning for both user and item sides. Meanwhile, HMRec achieves a steadier improved recommendation result for it has smaller variance in ten repeated experiments. Therefore, we can see the contribution of every component that complements each other in the proposed method HMRec.

We also analyze the effect of the key parameters α, β, γ through the ablation study as shown in Fig. 6. When $\alpha = 0$, the reconstruction loss of the source domains is dropped, in which case the source domain latent features may be unreasonable and even lead to a trivial solution. The loss for the dual-side domain discrimination task is dropped when $\beta = 0$, so the adversarial training will not be effective. When $\gamma = 0$, we drop the orthogonality constraint loss for the domain-specific and -shared latent features. Then the domain-specific feature subspace and the domain-shared feature subspace may be hard to separate. As shown in Fig. 6(a) and (b), the proposed HMRec method with the complete loss function achieves the best performance. Without any of the trade-off parameters may result in 1.75%–13.92% more errors, which reveals the necessity of each part of the loss function.

E. Parameters and Complexity Analysis

The most important parameter in the proposed method is the latent feature dimension in each part of the network. Fig. 7(a) shows the impact of the number of latent feature dimensions in HMRec. We choose the latent dimension that works the best for each network during the experiment process (e.g. the latent dimension of 20 works best for Task 1). The hyperparameter λ in the gradient reverse layer is set to be the constant 1 following the setting in [45].

We update the parameters using the target domain data and the source domain data in turn during training. The learning procedure is similar to DARec but is less time-consuming, because our methods are built in an end-to-end manner instead of being several separate structures [30]. The cost of the HMRec network for the training process of each domain is approximately equal to that of running a typical deep learning CF method [28]. The time and space complexity of the model is increased with the growth of the number of source domains,

but the marginal cost decreases. Overall, the entire network can be efficiently trained by backward propagation with minibatch Adam optimization. We show the experiment training loss (mean squared error) of Task 1 in Fig. 7(b) as an example.

VI. Conclusion

In this article, we propose a novel deep heterogeneous multidomain recommender system that achieves knowledge transfer through an adversarial learning process, named HMRec. It aims at solving two challenging issues: how to exploit the valuable information from multiple source domains when no single-source domain is sufficient as well as how to ensure positive transfer and avoid negative transfer when the data in different source domains are heterogeneous. Different loss functions and constraints are applied to effectively extract knowledge from both explicit numerical ratings and implicit binary interactions of multiple domains. Compared with existing works, HMRec can simultaneously extract domain-specific features for each domain as well as learn domain-shared features from multiple heterogeneous source domains to achieve better performance in cross-domain recommendation tasks. It shows great superiority compared with those methods that only rely on shared information from a single-source domain with the same kind of data. Extensive experiments on four tasks with real-world datasets under different sparsity ratios show that HMRec can outperform six state-of-the-art baselines in terms of RMSE and MAE.

Through the proposed method HMRec, we can maximally learn effective shared information from the available source domains to improve the performance of recommendation tasks in the target domain, which can ensure a positive transfer and avoid negative transfer to the greatest extent. However, there still are some interesting issues that are worthy to discuss. First, how to visualize the knowledge transfer process since the latent feature spaces in the recommender system is different from the feature spaces of images in most of the transfer learning methods. Second, how to select source domains and use less data to achieve better performance when possible. We still have a long way to go and this direction deserves further research.

For future work, we are developing a multidomain recommender system to provide users with personalized healthcare services with multiple source data from both genotypes and phenotypes. Our current work can be extended to more general scenarios where the side information of users or items is available. The multitask recommendation and the cross-domain recommendation tasks with time sequential data are also worth in-depth investigation.

Acknowledgment

This work was done during W. Liao's visit to the University of Technology Sydney.

References

[1] J. Lu, Q. Zhang, and G. Zhang, *Recommender Systems: Advanced Developments*. Singapore: World Scientific, 2020.

[2] S. Boutemedjet and D. Ziou, "Predictive approach for user long-term needs in content-based image suggestion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1242–1253, Aug. 2012.

[3] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. SIGKDD*, Aug. 2016, pp. 353–362.

[4] W. Wang, G. Zhang, and J. Lu, "Member contribution-based group recommender system," *Decis. Support Syst.*, vol. 87, pp. 80–93, Jul. 2016.

[5] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[7] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proc. IJCAI*, Aug. 2017, pp. 3203–3209.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. WWW*, 2017, pp. 173–182.

[9] S. Yu, X. Wang, W. Zhu, P. Cui, and J. Wang, "Disparity-preserved deep cross-platform association for cross-platform video recommendation," in *Proc. IJCAI*, Aug. 2019, pp. 4635–4641.

[10] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. WSDM*, Feb. 2017, pp. 425–434.

[11] M. Mao, J. Lu, G. Zhang, and J. Zhang, "Multirelational social recommendations via multigraph ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4049–4061, Dec. 2017.

[12] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. ICDM*, Nov. 2018, pp. 197–206.

[13] Q. Zhang, J. Lu, D. Wu, and G. Zhang, "A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1998–2012, Jul. 2019.

[14] Q. Zhang, W. Liao, G. Zhang, B. Yuan, and J. Lu, "A deep dual adversarial network for cross-domain recommendation," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 7, 2021, doi: 10.1109/TKDE.2021.3132953.

[15] F. Zhu, Y. Wang, C. Chen, G. Liu, M. Orgun, and J. Wu, "A deep framework for cross-domain and cross-system recommendations," in *Proc. IJCAI*, Jul. 2018, pp. 3711–3717.

[16] P. Li and A. Tuzhilin, "DDTCDR: Deep dual transfer cross domain recommendation," in *Proc. WSDM*, Jan. 2020, pp. 331–339.

[17] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," in *Proc. WWW*, 2013, pp. 595–606.

[18] B. Li, Q. Yang, and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model," in *Proc. ICML*, 2009, pp. 617–624.

[19] C. Wang, M. Niepert, and H. Li, "RecSys-DAN: Discriminative adversarial networks for cross-domain recommender systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2731–2740, Aug. 2020.

[20] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. SIGKDD*, 2008, pp. 650–658.

[21] Q. Zhang, D. Wu, J. Lu, F. Liu, and G. Zhang, "A cross-domain recommender system with consistent information transfer," *Decis. Support Syst.*, vol. 104, pp. 49–63, Dec. 2017.

[22] B. Loni, Y. Shi, M. Larson, and A. Hanjalic, "Cross-domain collaborative filtering with factorization machines," in *Proc. ECIR*, 2014, pp. 656–661.

[23] W. Pan *et al.*, "Transfer to rank for heterogeneous one-class collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 37, no. 1, pp. 1–20, Jan. 2019.

[24] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[25] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction," in *Proc. IJCAI*, vol. 9, 2009, pp. 2052–2057.

[26] S. Jiang, Z. Ding, and Y. Fu, "Deep low-rank sparse collective factorization for cross-domain recommendation," in *Proc. MM*, Oct. 2017, pp. 163–171.

[27] S. Jiang, Z. Ding, and Y. Fu, "Heterogeneous recommendation via deep low-rank sparse collective factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1097–1111, May 2020.

[28] G. Hu, Y. Zhang, and Q. Yang, "CoNet: Collaborative cross networks for cross-domain recommendation," in *Proc. CIKM*, Oct. 2018, pp. 667–676.

[29] J. He, R. Liu, F. Zhuang, F. Lin, C. Niu, and Q. He, "A general cross-domain recommendation framework via Bayesian neural network," in *Proc. ICDM*, Nov. 2018, pp. 1001–1006.

[30] F. Yuan, L. Yao, and B. Benatallah, "DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns," in *Proc. IJCAI*, Aug. 2019, pp. 4227–4233.

[31] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[32] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. CVPR*, Jul. 2017, pp. 7167–7176.

[33] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. NIPS*, 2016, pp. 343–351.

[34] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[35] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. AAAI*, 2018, pp. 3934–3941.

[36] I. Durugkar, I. Gemp, and S. Mahadevan, "Generative multi-adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.

[37] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[38] J. Lu, H. Zuo, and G. Zhang, "Fuzzy multiple-source transfer learning," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 12, pp. 3418–3431, Dec. 2020.

[39] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl.-Based Syst.*, vol. 80, pp. 14–23, May 2015.

[40] B. Tan, E. Zhong, E. W. Xiang, and Q. Yang, "Multi-transfer: Transfer learning with multiple views and multiple sources," in *Proc. SDM*, May 2013, pp. 243–251.

[41] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.

[42] Y.-H.-H. Tsai, Y.-R. Yeh, and Y.-C.-F. Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proc. CVPR*, Jun. 2016, pp. 5081–5090.

[43] F. Liu, G. Zhang, and J. Lu, "Heterogeneous domain adaptation: An unsupervised approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5588–5602, Dec. 2020.

[44] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. WWW*, May 2015, pp. 111–112.

[45] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.

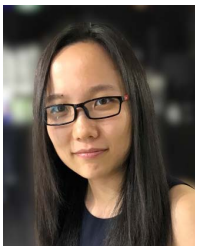
[46] W. Pan and Q. Yang, "Transfer learning in heterogeneous collaborative filtering domains," *Artif. Intell.*, vol. 197, pp. 39–55, Apr. 2013.

[47] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. WWW*, 2018, pp. 689–698.

[48] J. Lu, J. Xuan, G. Zhang, and X. Luo, "Structural property-aware multilayer network embedding for latent factor analysis," *Pattern Recognit.*, vol. 76, pp. 228–241, Apr. 2018.

Wenhui Liao received the B.Sc. degree in information and computing science from the School of Mathematics and Statistics, Shandong University, China, in 2018. She is currently pursuing the master's degree with the Shenzhen International Graduate School, Tsinghua University.

From 2019 to 2020, she was a Visiting Student with the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Australian Artificial Intelligence Institute (AAII), Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. Her research interests include recommender systems and transfer learning techniques, especially in cross-domain recommender systems.



Qian Zhang (Member, IEEE) received the Ph.D. degree from the University of Technology Sydney, Ultimo, NSW, Australia, in 2018.

She is currently a Post-Doctoral Research Fellow and a member of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Australian Artificial Intelligence Institute (AAII), Faculty of Engineering and Information Technology, University of Technology Sydney. Her research interests include recommender systems and personalized techniques. She specializes in cross-domain recommender systems.

Bo Yuan (Senior Member, IEEE) received the B.E. degree from the Nanjing University of Science and Technology, Nanjing, China, in 1998, and the M.Sc. and Ph.D. degrees from The University of Queensland (UQ), Queensland, NSW, Australia, in 2002 and 2006, respectively, all in computer science.

From 2006 to 2007, he was a Research Officer on a project funded by the Australian Research Council at UQ. He is currently an Associate Professor with the Division of Informatics, Shenzhen International Graduate School, Tsinghua University, Shenzhen, China, and a member of the Intelligent Computing Laboratory. His research interests include Intelligent Computing and Pattern Recognition.

Guangquan Zhang received the Ph.D. degree in applied mathematics from Curtin University, Bentley, WA, Australia, in 2001.

From 1993 to 1997, he was a full Professor with the Department of Mathematics, Hebei University, Baoding, China. He is currently an Australian Research Council (ARC) QEII Fellow, an Associate Professor and the Director of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory, Australian Artificial Intelligence Institute (AAII), University of Technology Sydney, Ultimo, NSW, Australia. His main research interests lie in the area of fuzzy machine learning, multiobjective, bilevel, and group decision making, and fuzzy measure. He has published six authored monographs, five edited research books, and over 450 articles including 250 refereed journal articles.

Dr. Zhang has won ten ARC Discovery Project grants, one ARC Linkage Project, and many other research grants and supervised 30 PhD students. He has served as a Guest Editor for five special issues of IEEE Transactions and other international journals.

Jie Lu (Fellow, IEEE) received the Ph.D. degree from Curtin University, Bentley, WA, Australia, in 2000.

She is currently a Distinguished Professor and the Director of Australian Artificial Intelligence Institute (AAII), University of Technology Sydney, Ultimo, NSW, Australia. She is also an IFSA Fellow and Australian Laureate Fellow. She has been awarded 10 Australian Research Council (ARC) discovery grants and led 15 industry projects. She has published over 500 articles in IEEE Transactions and other journals and conferences, supervised 50 Ph.D. students to completion. Her main research expertise is in transfer learning, concept drift, decision support systems, and recommender systems.

Dr. Lu serves as an Editor-In-Chief for Knowledge-Based Systems (Elsevier) and an Editor-In-Chief for *International Journal on Computational Intelligence Systems* (Atlantis). She has delivered 30 keynote speeches at IEEE and other international conferences and chaired 20 international conferences. She has received many national and international awards, including the UTS Medal for research excellence in 2019, the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award in 2019, the Computer Journal Wilkes Award in 2018, and the Australian Most Innovative Engineer Award in 2019.