Li Y, Xu JJ, Zhao PP *et al.* ATLRec: An attentional adversarial transfer learning network for cross-domain recommendation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(4): 794–808 July 2020. DOI 10.1007/s11390-020-0314-8

ATLRec: An Attentional Adversarial Transfer Learning Network for Cross-Domain Recommendation

Ying Li, *Member, CCF*, Jia-Jie Xu*, *Member, CCF, ACM*, Peng-Peng Zhao, *Member, CCF, ACM, IEEE*
Jun-Hua Fang, Wei Chen, *Member, CCF*, and Lei Zhao, *Member, CCF, ACM*

School of Computer Science and Technology, Soochow University, Suzhou 215006, China

E-mail: 20184227011@stu.suda.edu.cn; {xujj, ppzhao, jhfang, 20124227003, zhaol}@suda.edu.cn

Received January 18, 2020; revised June 3, 2020.

Abstract Entity linking is a new technique in recommender systems to link users' interaction behaviors in different domains, for the purpose of improving the performance of the recommendation task. Linking-based cross-domain recommendation aims to alleviate the data sparse problem by utilizing the domain-sharable knowledge from auxiliary domains. However, existing methods fail to prevent domain-specific features to be transferred, resulting in suboptimal results. In this paper, we aim to address this issue by proposing an adversarial transfer learning based model ATLRec, which effectively captures domain-sharable features for cross-domain recommendation. In ATLRec, we leverage adversarial learning to generate representations of user-item interactions in both the source and the target domains, such that the discriminator cannot identify which domain they belong to, for the purpose of obtaining domain-sharable features. Meanwhile each domain learns its domain-specific features by a private feature extractor. The recommendation of each domain considers both domain-specific and domain-sharable features. We further adopt an attention mechanism to learn item latent factors of both domains by utilizing the shared users with interaction history, so that the representations of all items can be learned sufficiently in a shared space, even when few or even no items are shared by different domains. By this method, we can represent all items from the source and the target domains in a shared space, for the purpose of better linking items in different domains and capturing cross-domain item-item relatedness to facilitate the learning of domain-sharable knowledge. The proposed model is evaluated on various real-world datasets and demonstrated to outperform several state-of-the-art single-domain and cross-domain recommendation methods in terms of recommendation accuracy.

Keywords adversarial transfer learning, attention mechanism, cross-domain recommendation, entity linking

1 Introduction

With the development of the Internet, users are more inclined to express their likes and dislikes for items on the Internet. As a result, users may have interaction records in different domains. Through entity linking, users' interaction behaviors in different domains can be linked together. Considering the problem of data sparsity and cold-start that single domain [1, 2] typically encounters, the linking of users in different domains opens up new opportunities to implement recommendation tasks more effectively and accurately. In real life, the same user is likely to have similar tastes across different domains, making entity linking a promising and practical technique for recommendation. It has been proven in [3–5] that the knowledge of a user in auxiliary domains contributes to enhancing the recommendation to her/him in the target domain. However, because of the existence of negative transfer, we face severe challenges in linking users' interaction behaviors in different domains, and making full use of the interactions in the auxiliary domains for the recommendation task in the target domain.

The existing linking-based approaches can be divided into: content-based, transfer-based, and embedding-based. Content-based approaches mainly focus

Regular Paper

Special Section on Entity Resolution

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61872258, 61772356, 61876117, and 61802273, and the Priority Academic Program Development of Jiangsu Higher Education Institutions of China.

*Corresponding Author

on linking different domains by identifying auxiliary contents such as user/item attributes [6], social tags [7], and user-generated texts [8]. Transfer-based approaches tend to transfer knowledge across domains by employing machine learning techniques, such as neural networks [9, 10] and transfer learning [11]. However, most existing transfer-based approaches pay more attention to how to transfer domain-sharable features than to how to obtain them, which results in domain-specific features being transferred. If domain-specific features are transferred from the source domain to the target domain, the source domain will introduce noise into the training of the target domain, resulting in worse recommendation performance in the target domain. In [12], the low-rank and sparse cross-domain recommendation model (LSCD) is proposed to solve the above problem to some extent. The method is based on cross-domain collaborative filtering (CDCF) [13], which transfers the knowledge of rating data among multiple domains and separates the user latent feature matrix into sharable and domain-specific parts adaptively to make a better recommendation. However, LSCD is a shallow model, which cannot learn complex nonlinear user-item interaction relationships. Embedding-based approaches focus on enhancing user or item modeling by means of network embedding and other embedding methods [14–16]. However, existing embedding-based methods rely on auxiliary contents such as textual data and visual data.

Adversarial transfer learning [17] is a technique that incorporates adversarial learning inspired by generative adversarial nets (GAN) [18] into transfer learning to find transferable knowledge which applies to both the source domain and the target domain. However, adversarial transfer learning ignores extracting domain-specific features in each domain which is also important for cross-domain recommendation. Though some approaches inspired by GAN for specific cross-domain recommendation scenarios have been proposed, they either ignore domain-specific features in each domain [19] or recommend items in the scenario where the target domain has no labeled data and requires auxiliary contents (e.g., reviews) [20, 21]. There is little related work to exploit adversarial transfer learning to train cross-domain recommendation task based on implicit feedback from both the source and the target domain, especially without auxiliary contents.

Different from the existing transfer-based approaches, we propose an adversarial transfer learning based model ATLRec. In ATLRec, we leverage the shared feature extractor to generate representations of user-item interactions in both domains, such that the domain discriminator cannot identify which domain they belong to, for the purpose of obtaining domain-sharable features. Meanwhile we use the private feature extractor to extract domain-specific features for each domain. The recommendation of each domain considers both domain-specific and domain-sharable features by a linear combination. For the private and the shared feature extractor, we choose the structure similar to the Multi-Layer Perception (MLP) model in [22]. We further adopt an attention mechanism to learn item latent factors of both domains by utilizing the shared users with interaction history, so that the representations of all items can be learned in a shared space, even when few or even no items are shared in different domains. By doing so, we can represent all items from the source domain and the target domain in a shared space, for the purpose of better linking items in different domains and capturing cross-domain item-item relatedness to facilitate the learning of domain-sharable knowledge.

We summarize the contributions of this paper as follows.

- We propose an adversarial transfer learning based model ATLRec for cross-domain recommendation, which effectively captures domain-sharable features to be transferred, using only the information from implicit feedback of both domains. Meanwhile the domain-specific features are also learned for each domain to be combined with domain-sharable features, for the purpose of learning comprehensive representations.

- For the purpose of better linking items in different domains and capturing cross-domain item-item relatedness to facilitate the learning of domain-sharable knowledge, we further adopt an attention mechanism to learn item latent factors of both domains by utilizing the shared users with interaction history, so that all items can be sufficiently represented in a shared space, even when few or even no items are shared in different domains.

- We conduct extensive experiments on various real-world datasets. Compared with some state-of-the-art single-domain and cross-domain recommendation methods, our model is demonstrated to be effective.

The following sections of this paper are organized as follows. We firstly introduce the related work in Section 2. In Section 3, we provide the notations used in this paper and describe the problem definition. We introduce the proposed framework of our model ATLRec in detail in Section 4. Section 5 shows our experimental results with analysis. Finally, we conclude the paper in Section 6.

2 Related Work

In this section, we briefly review the work related to cross-domain linking-based recommendation, including deep neural network and adversarial network techniques employed for it. In addition, we discuss attention-based techniques used in recommendation algorithms.

Cross-domain linking-based recommendation can be divided into: content-based, transfer-based, and embedding-based. Content-based approaches mainly focus on linking different domains by identifying auxiliary contents. The work proposed in [23] targets the data sparsity problem by combining user preferences and extracting common attributes of both users and items. Later on, the approach in [6] was proposed which can obtain accurate modeling of users' interest and needs by incorporating auxiliary contents from other systems. The work in [7] utilizes social tags to link domains, while the work in [8] uses user-generated texts to do the same thing.

Transfer-based approaches transfer knowledge across domains by employing machine learning techniques. Especially, a class of these methods are based on matrix factorization (MF) applied to each domain. Collective matrix factorization (CMF)[24] jointly factorizes rating matrices in the source domain and the target domain by sharing latent factors of users. The work in [3] proposes a codebook method to transfer user-item rating patterns from auxiliary domains to the target domain with a sparse rating matrix. LSCD[12] based on CDCF[13] transfers the knowledge of rating data among multiple domains and separates the user latent feature matrix into sharable and domain-specific parts adaptively to make a better recommendation. The work proposed in [9] and the work in [10] use a deep neural network to learn nonlinear mapping function between the source domain and the target domain. However, the above methods have difficulty in learning highly nonlinear user-item interaction relationships[22]. The work in [11] proposes collaborative cross-networks (CoNet) to learn complex user-item interaction relationships and transfer knowledge across domains by using neural networks as the base model. However, these neural network based approaches pay more attention to how to transfer domain-sharable features than to how to obtain them, which results in domain-specific features being transferred. Meanwhile cross-domain item-item relatedness is usually not considered in these approaches.

Embedding-based approaches integrate knowledge from various networks to enhance user or item modeling by means of network embedding and other embedding methods. This kind of problems are difficult to solve with traditional machine learning methods due to their complex structures. For example in [14], the auxiliary domain is a knowledge base with heterogeneous information, including both structured and unstructured data. The authors in [14] applied embedding methods including heterogeneous network embedding and deep learning embedding to automatically extract items' structural, textual, and visual representations from the knowledge base. The work proposed in [15] focuses on effective user embedding by using heterogeneous network embedding to jointly learn users' related information across multiple heterogeneous social networks. However, these above methods rely on auxiliary contents such as textual data, visual data, and content words.

In recent years, adversarial networks have achieved great success in computer vision[18, 25] and natural language processing (NLP) area[26, 27]. In cross-domain recommendation, adversarial networks have been introduced for domain adaptation. RecSys-DAN[20] learns to adapt domain indistinguishable representations in various modalities such as images and reviews from the source domain with the labeled data to the target domain without the labeled data in an unsupervised and adversarial fashion. The work proposed in [21] proposes the domain separation network to recommend items in the source domain to users in the target domain in an unsupervised mode and in need of auxiliary contents. The work in [19] proposes a deep domain adaption model to extract and transfer domain-sharable patterns from rating matrices only, but ignores the domain-specific features.

Attention mechanism[28, 29] has also shown great potential in recommendation algorithms. The work in [30] proposes attentive collaborative filtering (ACF) to introduce both component-level and item-level attention for multimedia recommendation into a collaborative filtering (CF) framework. The authors in [31] utilized attention mechanism to capture the varying attention that a user pays to each aspect of different items. The work proposed in [32] uses attention mechanism to control the ratio of the content-based (CB) and CF information for each user-item pair when making recommendations. The work in [33] introduces the BP neural network with attention mechanism (BPAM) which designs an attention mechanism to capture the global impact of the target user's neighbors by means of introducing their global weights. However few attention mechanism techniques are used for cross-domain recommendation.

Different from the above cross-domain methods, our approach can effectively capture domain-sharable features to be transferred and domain-specific features for each domain, using only the information from the implicit feedback of both domains. In addition, motivated by the above attention-based models, we adopt an attention mechanism to represent all items in a shared space, for the purpose of better linking items in different domains and capturing cross-domain item-item relatedness to facilitate the learning of domain-sharable knowledge.

3 Notations and Problem Definition

In this section, we will introduce the notations used in this paper and describe the problem definition. For the sake of understanding, we list the notations in Table 1.

We are given a source domain $\mathcal{S}$ and a target domain $\mathcal{T}$, where the set of users $U = \{u_1, ..., u_m\}$ (of size $m = |U|$) in both domains is shared. We denote the set of items in the source domain $\mathcal{S}$ and the target domain $\mathcal{T}$ by $I_{\rm S} = \{i_1, ..., i_{n_{\rm S}}\}$ and $I_{\rm T} = \{j_1, ..., j_{n_{\rm T}}\}$ (of size $n_{\rm S} = |I_{\rm S}|$ and $n_{\rm T} = |I_{\rm T}|$) respectively. In this paper, we implement the recommendation task with implicit feedback [34, 35]. Therefore, we use matrix $\boldsymbol{R}_{\rm T} \in \mathbb{R}^{m \times n_{\rm T}}$ to represent the user-item interaction matrix in the target domain, where the entry $r_{uj} \in \{0, 1\}$ is 1 (observed) if user u has an interaction with item j and 0 (unobserved) otherwise. Similarly, let another binary matrix $\boldsymbol{R}_{\rm S} \in \mathbb{R}^{m \times n_{\rm S}}$ denote user-item interactions in the source domain, and the entry $r_{ui} \in \{0, 1\}$ is 1 if user u has an interaction with item i and 0 otherwise. We denote the set of observed items given by user u as $I_u^{\rm S}$ ($I_u^{\rm T}$), and the unobserved items as $\overline{I}_u^{\rm S}$ ($\overline{I}_u^{\rm T}$) in the source (target) domain. Similarly, let U_i (U_j) denote the users who have interactions with item i (j) from the source (target) domain and $\overline{U}_i$ ($\overline{U}_j$) otherwise.

Problem Definition. Given two observed domains including the user-item interaction matrices $\boldsymbol{R}_{\rm S}$ and $\boldsymbol{R}_{\rm T}$, our goal for the cross-domain recommendation task is to recommend the items from $\overline{I}_u^{\rm T}$ to user u by linking users' interaction behaviors in both $\boldsymbol{R}_{\rm S}$ and $\boldsymbol{R}_{\rm T}$.

4 Proposed Framework

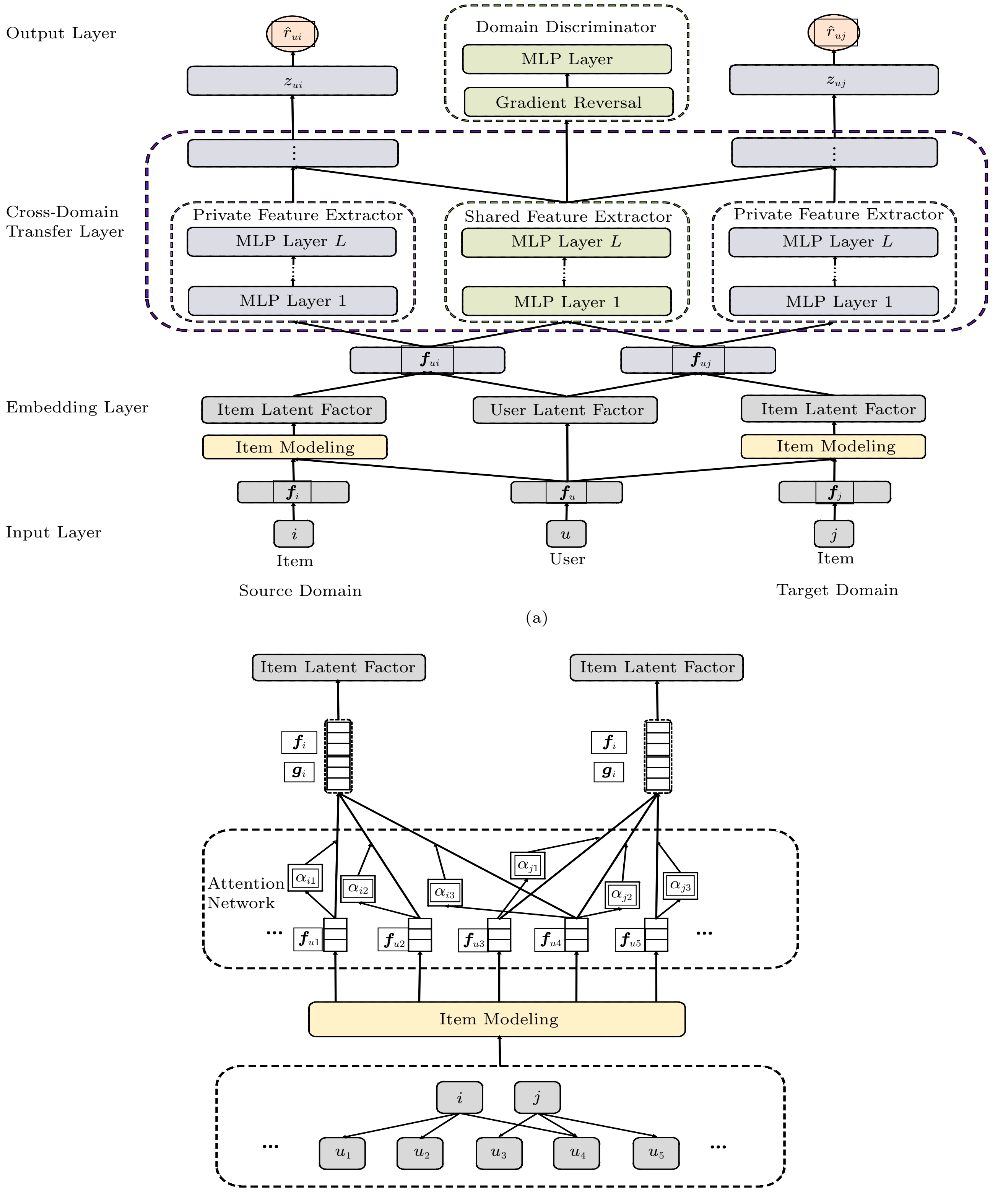
In this section, we first give an overview of the proposed framework, and then the details of each model component will be introduced. And finally we discuss how to learn the model parameters.

4.1 Overview of Proposed Framework

The architecture of the proposed approach is shown in Fig.1. The model mainly consists of four components: an embedding layer with item modeling, a cross-domain transfer layer composed of the shared-private feature extractor, the cross-domain recommendation part, and the domain discriminator. The first component is the embedding layer with item modeling, which is to learn latent factors of users and items. Especially for each item i (j) in item modeling, we introduce an attention mechanism to aggregate the in-

Table 1. Definitions of Notations

Notation	Definition
$\mathcal{S}$, $\mathcal{T}$	Source domain and target domain
U, m	Shared users of both domains $U = \{u_1, ..., u_m\}$ and its size $m = \|U\|$
$I_{\rm S}$, $n_{\rm S}$	Set of items $I_{\rm S} = \{i_1, ..., i_{n_{\rm S}}\}$ in the source domain and its size $n_{\rm S} = \|I_{\rm S}\|$
$I_{\rm T}$, $n_{\rm T}$	Set of items $I_{\rm T} = \{j_1, ..., j_{n_{\rm T}}\}$ in the target domain and its size $n_{\rm T} = \|I_{\rm T}\|$
$\boldsymbol{R}_{\rm S} \in \mathbb{R}^{m \times n_{\rm S}}$, $\boldsymbol{R}_{\rm T} \in \mathbb{R}^{m \times n_{\rm T}}$	User-item interaction matrix in the source and the target domain
$r_{ui} \in \{0, 1\}$	Entry in the source domain user-item interaction matrix, where 1 represents user u has an interaction with the item i in the source domain and 0 otherwise
$r_{uj} \in \{0, 1\}$	Entry in the target domain user-item interaction matrix, where 1 represents user u has an interaction with the item j in the target domain and 0 otherwise
$I_u^{\rm S}$, $\overline{I}_u^{\rm S}$	Set of observed and unobserved items given by user u in the source domain
$I_u^{\rm T}$, $\overline{I}_u^{\rm T}$	Set of observed and unobserved items given by user u in the target domain
U_i, $\overline{U}_i$	Users who have and do not have interactions with item i in the source domain
U_j, $\overline{U}_j$	Users who have and do not have interactions with item j in the target domain



(b)

Fig.1. General architecture of our proposed model. (a) Proposed adversarial transfer learning based model architecture for cross-domain recommendation. The left and the right are the source and the target domain respectively, and the middle part is the shared space consisting of the shared feature extractor and the domain discriminator. The shared space attempts to learn domain-sharable features from the source and the target domain and then transfer them for the purpose of improving the target domain recommendation performance. (b) Attention network architecture for item modeling.

formation from the set of users who have interactions with i (j), denoted as U_i (U_j). The second component is the shared-private feature extractor, which attempts to learn domain-specific and domain-sharable features by adopting the structure similar to the MLP model in [22]. The third component is the cross-domain recommendation part, which utilizes the MLP model and considers both domain-sharable and domain-specific features, for the purpose of the task of item recommendation. In the domain discriminator component, we incorporate adversarial learning to prevent domain-specific features to be transferred. In Subsection 4.2–Subsection4.6, we will introduce our model in detail.

4.2 Embedding Layer

4.2.1 Embedding

We use one-hot encoding to encode user-item interaction indices for the purpose of representing the input. For user u, item i in the source domain, and item j in the target domain, we map them into one-hot encodings $\boldsymbol{x}_u \in \{0,1\}^m$, $\boldsymbol{x}_i \in \{0,1\}^{n_{\mathrm{S}}}$ and $\boldsymbol{x}_j \in \{0,1\}^{n_{\mathrm{T}}}$ to represent the vectors consisting of 0 and 1, where only the element corresponding to the index is 1 and the others are 0. Then we embed one-hot encodings into continuous representations $\boldsymbol{f}_u = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_u$, $\boldsymbol{f}_i = \boldsymbol{Q}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{x}_i$ and $\boldsymbol{f}_j = \boldsymbol{Q}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{x}_j$ by embedding matrices $\boldsymbol{P}$, $\boldsymbol{Q}_{\mathrm{s}}$ and $\boldsymbol{Q}_{\mathrm{t}}$ respectively. For items i and j in the source domain and the target domain, we further use item modeling to learn item latent factors $\boldsymbol{F}_i$ and $\boldsymbol{F}_j$. Then we merge them as $\boldsymbol{f}_{ui} = [\boldsymbol{f}_u, \boldsymbol{F}_i]$ and $\boldsymbol{f}_{uj} = [\boldsymbol{f}_u, \boldsymbol{F}_j]$ to be the input of successive layers.

4.2.2 Item Modeling

As shown in Fig.1(b), item modeling is used to learn item latent factor by aggregating users. For each item j (i) in the target (source) domain, we need to aggregate the information from the set of users who have interactions with j (i), denoted as U_j (U_i). Considering that users are shared, items in different domains can be mapped to the same representation space and be represented sufficiently.

We take item j in the target domain, the item latent factor of which is denoted by $\boldsymbol{F}_j$ as an example. The same is true for each item i in the source domain. The aggregation function here is denoted by $Aggre$.

$$\boldsymbol{g}_j = Aggre(\{\boldsymbol{f}_{u_j}, \forall u_j \in U_j\}).$$

Considering that different users who have interacted with the same item may have different informativeness in representing this item, we introduce an attention mechanism to model the importance of the users that connect to item j, which is formulated as:

$$\begin{aligned} \boldsymbol{g}_j &= \sum_{u_j \in U_j} \alpha_{u_j} \boldsymbol{f}_{u_j}, \\ \alpha^*_{u_j} &= \boldsymbol{w}_2^{\mathrm{T}} \cdot \sigma(\boldsymbol{W}_1 \cdot \boldsymbol{f}_{u_j} + \boldsymbol{b}_1) + b_2, \\ \alpha_{u_j} &= \frac{\exp(\alpha^*_{u_j})}{\sum_{u_j \in U_j} \exp(\alpha^*_{u_j})}, \end{aligned}$$

where $\boldsymbol{W}_1$, $\boldsymbol{b}_1$, $\boldsymbol{w}_2$ and b_2 are parameters in this network. σ is the activation function and we choose Leaky ReLU here.

Since some latent properties of item j may not be revealed by the users with interaction history, we use the MLP model, denoted by MLP, to fuse the information of the item ID embedding $\boldsymbol{f}_j$ and $\boldsymbol{g}_j$, for the purpose of enhancing the representation, which is formulated as:

$$\boldsymbol{F}_j = MLP([\boldsymbol{f}_j \oplus \boldsymbol{g}_j]),$$

where $\oplus$ denotes concatenation operation.

4.3 Shared-Private Feature Extractor

For the shared-private feature extractor, we adopt the MLP model to learn interaction features between user and item latent factor. The MLP model is defined as:

$$\begin{aligned} &MLP(\boldsymbol{x}|\theta) = \phi_L(\phi_{L-1}(...(\phi_1(\boldsymbol{x}))...)), \\ &\phi_L(\boldsymbol{x}) = \sigma_L(\boldsymbol{W}_L^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{b}_L), \end{aligned}$$

where $\boldsymbol{W}_L$, $\boldsymbol{b}_L$, σ_L denote the weight matrix, the bias vector, and the activation function for the L-th layer respectively and θ denotes all the parameters for the MLP model. We choose Leaky ReLU for σ_L here.

The shared-private feature extractor includes private MLP layers for the specific domain and shared MLP layers for both domains. The private MLP layers are used to extract domain-specific features to be transformed and the shared MLP layers are used to extract domain-sharable features to be transferred. Formally, the shared and the private MLP layers can be expressed as follows:

$$\begin{aligned} \boldsymbol{f}_{\mathrm{T}}^{\mathrm{s}} &= MLP(\boldsymbol{f}_{ui}|\theta_{\mathrm{shared}}), \\ \boldsymbol{f}_{\mathrm{T}}^{\mathrm{p}} &= MLP(\boldsymbol{f}_{uj}|\theta_{\mathrm{T}}), \\ \boldsymbol{f}_{\mathrm{S}}^{\mathrm{s}} &= MLP(\boldsymbol{f}_{uj}|\theta_{\mathrm{shared}}), \\ \boldsymbol{f}_{\mathrm{S}}^{\mathrm{p}} &= MLP(\boldsymbol{f}_{ui}|\theta_{\mathrm{S}}), \end{aligned}$$

where θ_{shared}, θ_{T} and θ_{S} are the shared MLP parameters, MLP parameters of the target domain, and MLP parameters of the source domain respectively.

4.4 Cross-Domain Recommendation

After extracting domain-sharable and domain-specific features from the shared-private feature extractor, we obtain a final latent representation $\boldsymbol{z}_{uj}$ by combining them:

$$\boldsymbol{z}_{uj} = MLP(\boldsymbol{f}_{\mathrm{T}}),$$
$$\boldsymbol{f}_{\mathrm{T}} = \sigma(\boldsymbol{W}_{\mathrm{T}}\boldsymbol{f}_{\mathrm{T}}^{\mathrm{p}} + \boldsymbol{H}_{\mathrm{T}}\boldsymbol{f}_{\mathrm{T}}^{\mathrm{s}}),$$

where $\boldsymbol{W}_{\mathrm{T}}$ denotes the transform matrix of domain-specific features $\boldsymbol{f}_{\mathrm{T}}^{\mathrm{p}}$ and $\boldsymbol{H}_{\mathrm{T}}$ denotes the transfer matrix of domain-sharable features $\boldsymbol{f}_{\mathrm{T}}^{\mathrm{s}}$ in the target domain. σ is the activation function and we use Leaky ReLU here. Then the items in the target domain are ranked by their predicted scores:

$$\hat{r}_{uj} = \sigma(\boldsymbol{z}_{uj}),$$

where we use softmax for σ here. And we predict scores $\hat{r}_{ui}$ of the items in the source domain in the same way.

4.5 Domain Discriminator

To guarantee that domain-specific features are not transferred, we propose a domain discriminator to estimate which domain the features come from, which is formulated as:

$$D(\boldsymbol{f}_k^{\mathrm{s}}|\theta_{\mathrm{D}}) = \sigma(MLP(\boldsymbol{f}_k^{\mathrm{s}}|\theta_{\mathrm{D}})),$$

where domain $k \in \{\mathrm{S}, \mathrm{T}\}$, θ_{D} denotes the parameters of the domain discriminator and we choose softmax for σ here.

We introduce an adversarial loss L_{Adv} to prevent domain-specific features to be transferred. The adversarial loss trains the shared feature extractor to learn domain-sharable features such that the domain discriminator cannot reliably estimate which domain the features belong to. The adversarial loss can be formulated as:

$$L_{\mathrm{Adv}}(\Theta_{\mathrm{Adv}}) = \min_{\theta_{\mathrm{shared}}}(\max_{\theta_{\mathrm{D}}} \sum_{k \in \{\mathrm{S},\mathrm{T}\}} \sum_{i=1}^{T_k} \log D(\boldsymbol{f}_k^{\mathrm{s}})),$$

where T_k is the number of training examples of domain k and Θ_{Adv} denotes parameters to be trained in the adversarial training. There is a minimax optimization that the shared feature extractor learns features to mislead the domain discriminator and the discriminator tries its best to correctly predict the corresponding domain label.

For the purpose of addressing the minimax optimization problem, we add a gradient reversal layer (GRL) [36]. We minimize the domain discriminator errors in the training phrase, and GRL will encourage the shared feature extractor to learn domain-sharable features in an adversarial way.

4.6 Model Learning

Because of the nature of implicit feedback and the task of item recommendation, we choose cross-entroy as our loss function:

$$\mathcal{L}_0 = -\sum_{(u,i)\in \boldsymbol{R}^+ \cup \boldsymbol{R}^-} r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log(1 - \hat{r}_{ui}),$$

where $\boldsymbol{R}^+$ and $\boldsymbol{R}^-$ are the observed interaction matrix and randomly sampled negative examples [35] respectively. The objective function has probabilistic interpretation and is the negative logarithm likelihood of the following likelihood function:

$$\mathcal{L}(\Theta|\boldsymbol{R}^+ \cup \boldsymbol{R}^-) = \prod_{(u,i)\in \boldsymbol{R}^+} \hat{r}_{ui} \prod_{(u,i)\in \boldsymbol{R}^-} (1 - \hat{r}_{ui}),$$

where Θ denotes model parameters. The final loss function of our proposed model can be written as follows:

$$\mathcal{L}(\Theta) = \mathcal{L}_{\mathrm{T}}(\Theta_{\mathrm{T}}) + \mathcal{L}_{\mathrm{S}}(\Theta_{\mathrm{S}}) + \lambda L_{\mathrm{Adv}}(\Theta_{\mathrm{Adv}}),$$

where λ is a hyper-parameter and the model parameters $\Theta = \Theta_{\mathrm{T}} \cup \Theta_{\mathrm{S}} \cup \Theta_{\mathrm{Adv}}$. This objective function can be optimized by stochastic gradient descent (SGD):

$$\Theta' \leftarrow \Theta - \eta \frac{\partial \mathcal{L}(\Theta)}{\partial \Theta},$$

where η is the learning rate.

5 Experiments

In this section, we first introduce the experimental settings and then experimental results are presented to validate our contributions.

5.1 Experimental Settings

In this subsection, we will introduce the datasets, the evaluation protocol, baselines, and implementation details.

5.1.1 Datasets

We evaluate on three real-world cross-domain datasets. The first is Mobile containing user-app installations and user-news reading records, which was released by [11]. The second dataset is Amazon with different domains [37], where we convert the ratings of

4–5 as positive examples and select users and items with at least five ratings. The last one is Douban crawled from the Douban website①. Similarly, we take the ratings of 4–5 as positive examples. The datasets and the statistics are summarized in Table 2.

5.1.2 Evaluation Protocol

We use the leave-one-out (LOO) evaluation in [22], which reserves one interaction as the test item for each user. We determine hyper-parameters by randomly sampling another interaction per user as the validation set. We follow the common strategy which randomly samples 99 negative items having no interactions with the user and then evaluate the ability of the recommender to rank the test item against the negative ones.

Aiming at top-N item recommendation, we choose the typical evaluation metrics: hit ratio (HR), normalized discounted cumulative gain (NDCG) and mean reciprocal rank (MRR), where the ranked list is cut off at $topN = 10$. HR intuitively measures whether the reserved test item is present on the top-N list, defined as:

$$HR = \frac{1}{|U|} \sum_{u \in U} \delta(p_u \leqslant topN),$$

where p_u is the hit position of the test item for user u and δ is the indicator function. NDCG and MRR represent the rank of the hit positions, which can be defined as:

$$NDCG = \frac{1}{|U|} \sum_{u \in U} \frac{\log 2}{\log(p_u + 1)},$$

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{p_u}.$$

The higher the value, the better the performance.

5.1.3 Baselines

We compare our model with the following baselines, including several state-of-the-art single-domain and cross-domain recommendation methods. The first two are single-domain methods while the others are cross-domain methods.

Bayesian Personalized Ranking (*BPR*)[38]. Bayesian personalized ranking is a typical CF approach, which learns the user and item latent factors via matrix factorization and pairwise rank loss.

CMF[24]. Collective matrix factorization is a multi-relation learning approach, which jointly factorizes user-item interaction matrices of both domains by sharing user latent factors here.

CDCF[13]. Cross-domain CF is a context-aware approach, which is based on FM and extends the single-domain feature vector to incorporate the information from auxiliary domains. It uses the auxiliary as context and applies factorization on the merged domains aligned by shared users.

LSCD[12]. LSCD is based on CDCF[13], which transfers the knowledge of rating data among multiple domains and separates the user latent feature matrix into sharable and domain-specific parts adaptively to make a better recommendation.

MLP[22]. Multi-layer perception is a typical neural collaborate filtering approach. It learns complex (non-linear) user-item interactions by using neural networks.

MLP++. MLP++ is a degenerated method with no transfer learning part, which combines two MLP models by sharing the user embedding matrix only.

Cross-Stitch Networks (*CSN*)[39]. The cross-stitch network is a deep multi-task learning model, which combines two domains linearly by a shared coefficient to transfer knowledge.

CoNet[11]. CoNet is a state-of-the-art deep cross-domain recommendation model. It learns complex user-item interaction relationships by using neural networks and enables dual knowledge transfer across domains by introducing improved cross connections based on the cross-stitch network.

Table 2. Datasets and Statistics

Dataset	Domain		Number of Items		Number of	Number of Interactions		Density	
	Source	Target	Source	Target	Shared	Source	Target	Source (%)	Target (%)
Mobile	News	App	29 921	14 348	23 111	617 146	1 164 394	0.089	0.351
Amazon	Movies and TV	Books	58 119	311 981	30 637	581 231	988 969	0.033	0.010
	Digital Music	Movies and TV	10 730	33 003	2 889	39 869	139 202	0.129	0.146
Douban	DoubanMovie	DoubanBook	22 411	5 909	2 002	893 561	85 781	1.992	0.725

Note: Source refers to the source domain. Target refers to the target domain. Shared refers to the shared users between domains. Mobile, Amazon and Douban are three real-world cross-domain datasets, each containing the source domain and the target domain.

①https://www.douban.com/, May 2020.

SCoNet[11]. SCoNet is an improved version of CoNet, which adaptively selects representations to transfer and has proven to perform better than CoNet.

5.1.4 Implementation

For BPR, we use the popular CF library LightFM②. For CMF, we use a Python version of the Matlab code③. For CDCF, we adopt the libFM implementation④. For MLP, CoNet and SCoNet, we adopt the codes provided by authors. For MLP++, we change the MLP code to combine two MLP models by sharing the user embedding matrix. For CSN, we make modifications based on the CoNet code, keeping the number of neurons in each hidden layer the same and turning the transfer matrix into a scalar. For each baseline, we try to get their best performance by adjusting parameters. Our model is implemented by TensorFlow and the parameters we use are initialized in a normal distribution $\mathcal{N}(0, 0.01^2)$. The optimizer we adopt is Adam with an initial learning rate of 0.001. The negative sampling ratio we adopt is 1 while the batch size is 128. For the shared-private feature extractor, we adopt two-layer MLP structure and the configuration is $[64 \rightarrow 64]$. As for the design of the MLP layer in the domain discriminator, the configuration of hidden layers is $[64 \rightarrow 32 \rightarrow 16 \rightarrow 2]$. For the rest of cross-domain recommendation, we use two-layer MLP structure with both domains configured at $[64 \rightarrow 64]$.

5.2 Comparison with Baselines

In order to prove the performance of our model ATLRec, we compare it with several state-of-the-art single-domain and cross-domain recommendation methods. The performance comparison results are illustrated in Table 3. To distinguish between two sets of data in Amazon without taking up space, we use Amazon(1) and Amazon(2) in Table 3 to represent the first and the second set of Amazon data in Table 2, respectively. The last column is the relative improvement compared with the best baselines. These baselines can be classified in two ways. Firstly, we can divide them into shallow models containing BPR, CMF, CDCF and LSCD and deep learning based models containing the rest. These methods can also be divided into single-domain methods including BPR and MLP and cross-domain methods including the rest. We can see that our proposed model performs better than all these baselines.

We can observe that deep learning based methods (e.g., MLP) perform better than shallow methods (e.g., BPR) in either single-domain or cross-domain, which proves complex nonlinear user-item interaction relationships learned by deep neural networks benefit not only single-domain but also cross-domain recommendation, especially in the case of extreme sparse data. For instance in the most sparse dataset, Amazon Movies & TV and Book, our model ATLRec improves more than 18% compared with shallow cross-domain models CMF, CDCF and LSCD in terms of all the three evaluation metrics.

Table 3. Performance Comparison of Multiple Methods

Dataset	Metric	BPR	CMF	CDCF	LSCD	MLP	MLP++	CSN	CoNet	SCoNet	ATLRec	Improvement (%)
Mobile	HR	0.640 1	0.791 1	0.790 5	0.829 7	0.842 5	0.843 1	0.845 1	0.845 9	0.847 6*	**0.855 6**	0.94
	NDCG	0.502 2	0.583 2	0.602 3	0.631 1	0.667 5	0.668 3	0.671 2	0.671 3	0.672 8*	**0.683 8**	1.63
	MRR	0.461 7	0.537 8	0.551 5	0.578 4	0.621 5	0.623 6	0.630 5	0.629 5	0.630 7*	**0.644 3**	2.16
Amazon(1)	HR	0.332 9	0.359 4	0.361 1	0.370 9	0.386 0	0.404 7	0.422 6	0.426 0	0.428 1*	**0.440 9**	2.99
	NDCG	0.220 7	0.230 5	0.221 9	0.238 8	0.251 5	0.252 3	0.265 1	0.275 2	0.280 5*	**0.301 9**	7.63
	MRR	0.174 6	0.190 6	0.183 4	0.230 6	0.209 9	0.260 0	0.269 5	0.281 1	0.293 1*	**0.311 6**	6.31
Amazon(2)	HR	0.351 9	0.389 1	0.385 6	0.401 7	0.422 6	0.423 3	0.424 4	0.438 2	0.440 3*	**0.455 9**	3.54
	NDCG	0.183 3	0.210 6	0.209 8	0.227 5	0.238 1	0.246 3	0.238 1	0.244 4	0.247 2*	**0.258 9**	4.73
	MRR	0.169 4	0.209 8	0.215 5	0.228 1	0.236 4	0.243 8	0.238 4	0.241 9	0.244 5*	**0.251 4**	2.82
Douban	HR	0.349 1	0.382 5	0.394 7	0.411 9	0.425 6	0.440 1	0.444 6*	0.439 6	0.442 5	**0.453 0**	1.89
	NDCG	0.160 3	0.184 4	0.206 9	0.231 2	0.247 5	0.266 3	0.265 6	0.261 1	0.266 5*	**0.273 5**	2.63
	MRR	0.156 9	0.169 4	0.173 7	0.210 8	0.193 0	0.264 6*	0.261 1	0.258 9	0.264 2	**0.269 1**	1.70

Note: The best model is boldfaced and the best baselines are followed by stars.

②https://github.com/lyst/lightfm, May 2020.
③http://www.cs.cmu.edu/~ajit/cmf/, May 2020.
④http://www.libfm.org/, May 2020.

The other thing we can see is that compared with single-domain methods, cross-domain methods perform better no matter in the case of deep learning based or shallow models, which proves the effectiveness of utilizing knowledge from auxiliary domains, especially under the data sparse problem. For instance in the two Amazon datasets, our model ATLRec achieves 14.22% and 7.88% improvements respectively in terms of HR compared with the deep learning based single-domain method MLP. However, different ways to utilize the source domain lead to different levels of performance improvement. In general, the last four methods including CSN, CoNet, SCoNet and our model ATLRec achieve better improvement than MLP++ sharing the user embedding matrix only based on shared users, which illustrates the importance of knowledge transfer. For instance in the Mobile dataset, ATLRec gains 1.48%, 2.32% and 3.32% improvements over MLP++ in terms of HR, NDCG and MRR, respectively.

Our model obtains obvious improvement over knowledge transfer based cross-domain methods CSN, CoNet and SCoNet. For instance in the Amazon Movies & TV and Books, ATLRec improves 2.99%, 7.63% and 6.31% in terms of HR, NDCG and MRR compared with the best baseline SCoNet. This result shows that what to transfer is a matter deserving great concern and transferring domain-sharable knowledge benefits the performance of recommender systems.

As we can see from what has been discussed above and Table 3, our model ATLRec performs better than all the baselines, which illustrate its effectiveness.

5.3 Sensitivity Analysis of λ

We analyse the sensitivity of the loss weight coefficient λ. We keep the structure of our proposed model and then change the coefficient λ from 0.05 to 1 for all datasets. From Fig.2, we can observe that these

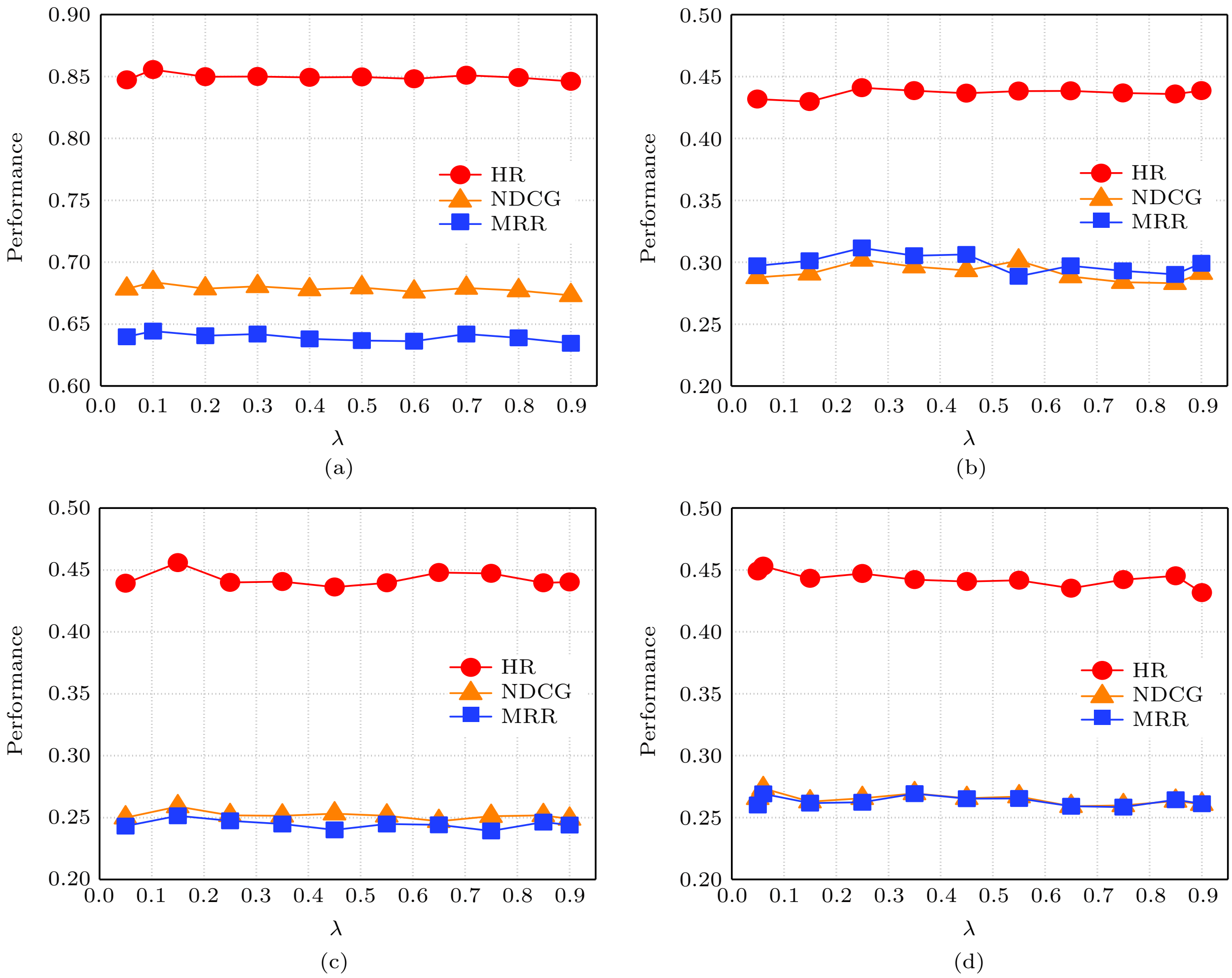


Fig.2. Sensitivity analysis of λ. (a) Dataset Mobile. (b) Dataset Amazon(1). (c) Dataset Amazon(2). (d) Dataset Douban.

datasets reach their good performance when λ is equal to 0.1, 0.25, 0.15 and 0.06, respectively. The loss weight coefficient λ represents a trade-off in extracting domain-sharable and domain-specific features. Therefore, the larger the optimal coefficient λ, the more the domain-sharable features need to be transferred.

5.4 Impact of Embedding Size

We investigate the impact of the embedding size on recommendation performance. We change the dimension of the embedding from 8 to 64 for all datasets, specifically 8, 16, 32, 50, 64. From Fig.3, it can be observed that these datasets reach their good performance when the embedding size is around 32, 32, 50 and 32, respectively. When the embedding dimension is too small, the features cannot be effectively extracted. When the size is too large, the extracted features will be too noisy.

5.5 Effectiveness of Adversarial Transfer Learning and Attention Mechanism

Table 4 shows the comparison of our proposed model with its simplified models on the HR test performance (NDCG and MRR have similar trends). The first model is the base MLP [22] model. We apply transfer learning to the base MLP model for the second model. In the third model, we incorporate adversarial training and consider both domain-specific and domain-sharable features. The last model is our proposed model, which incorporates adversarial training and attention mechanism.

From Table 4 we can observe that the performance is gradually improving with the addition of each part, which proves the effectiveness of adversarial transfer learning and attention mechanism. For instance in the Amazon Digital Music and Movies & TV, the second model improves 4.24% in terms of HR compared with the base MLP model, which demonstrates the effec-

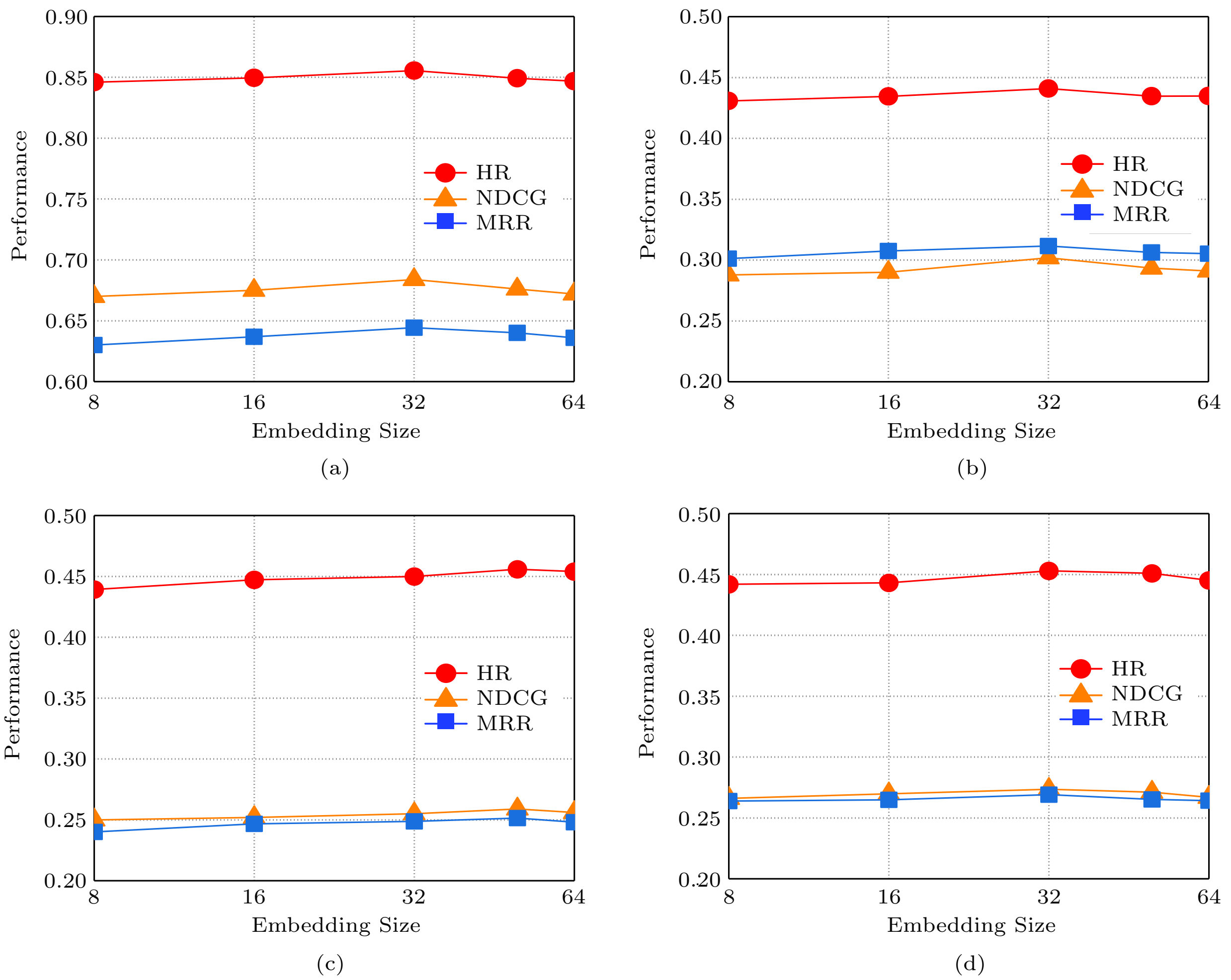


Fig.3. Impact of embedding size. (a) Dataset Mobile. (b) Dataset Amazon(1). (c) Dataset Amazon(2). (d) Dataset Douban.

tiveness of transfer learning. In the Douban dataset, the model incorporating adversarial training improves 1.67% compared with the model with transfer learning, which proves the effectiveness of adversarial transfer learning. In the Amazon Movies & TV and Books, our model improves 2.37% compared with the model without attention mechanism, which proves the necessity of attention mechanism.

Table 4. Performance Comparison Between Our Model and Simplified Models

Model	Mobile	Amazon(1)	Amazon(2)	Douban
MLP [22]	0.842 5	0.386 0	0.422 6	0.425 6
+Transfer	0.846 9	0.426 7	0.440 5	0.442 6
+Adversarial	0.852 9	0.430 7	0.447 2	0.450 0
+Adv+Attention	**0.855 6**	**0.440 9**	**0.455 9**	**0.453 0**

Note: The comparison is based on the HR test performance, and the best model is boldfaced.

5.6 Optimization Performance

We analyze the optimization performance of our model ATLRec in different training stages. Figs.4(a)–4(d) show the training loss and NDCG test performance in four datasets (HR and MRR have similar trends) varying with training epochs, respectively. We can observe that the training loss gradually decreases and the recommendation performance improves accordingly with more iterations. In the Mobile dataset, the most effective updates occur in the first 20 iterations, while the most effective updates occur in the first 30 iterations for the other datasets. Fig.4 also shows that our model is relatively stable with more iterations.

6 Conclusions

In this paper, we proposed an adversarial transfer learning based model ATLRec to link users' interaction

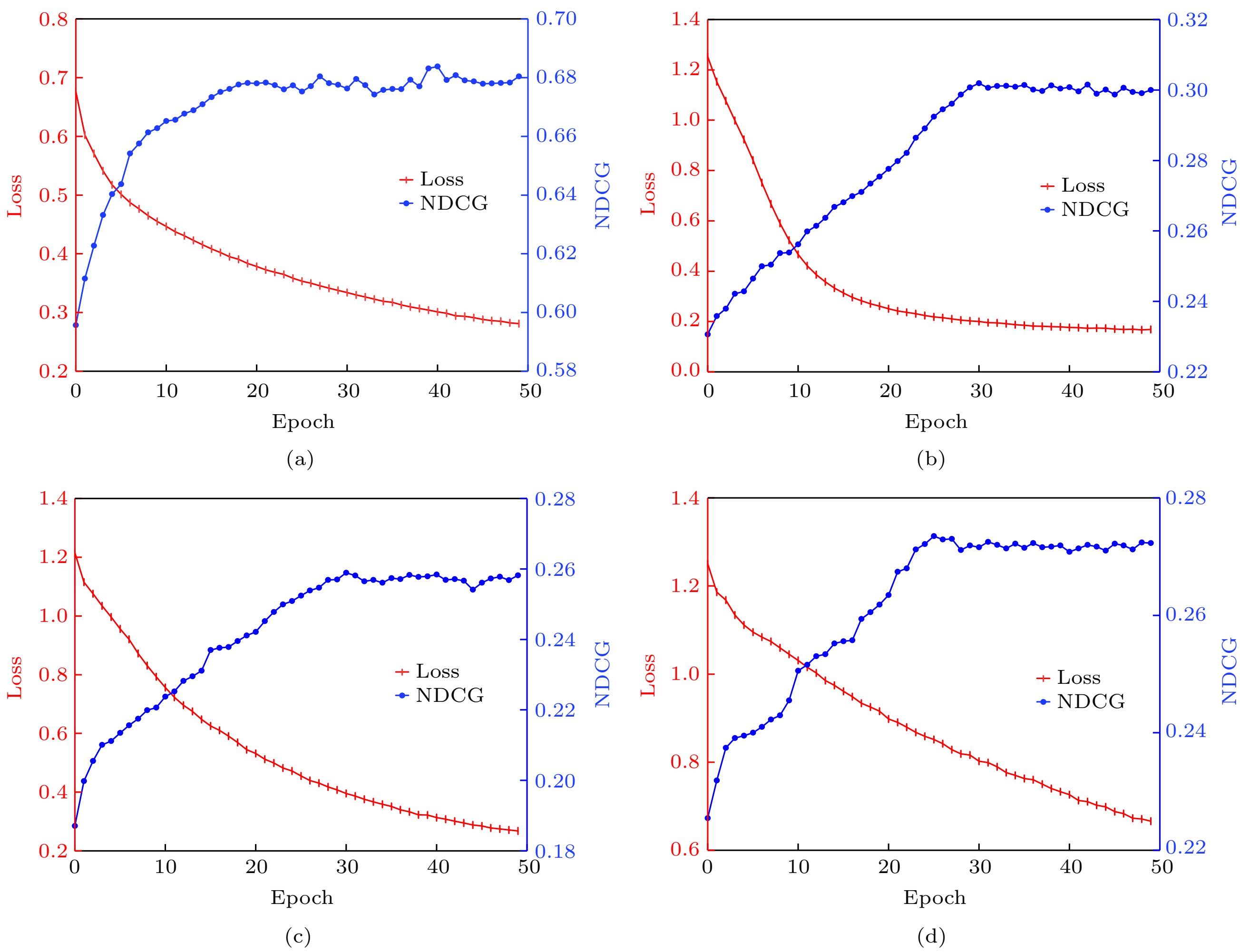


Fig.4. Loss and performance. (a) Dataset Mobile. (b) Dataset Amazon(1). (c) Dataset Amazon(2). (d) Dataset Douban.

behaviors in different domains for cross-domain recommendation, which can effectively capture domain-sharable features and domain-specific features at the same time, for the purpose of learning comprehensive representations. In addition, we adopted an attention mechanism to learn their item latent factors by utilizing the shared users with an interaction history, for the purpose of better linking items in different domains and capturing cross-domain item-item relatedness to facilitate the learning of domain-sharable knowledge, even when few or even no items are shared by different domains. The superior performance of our model has been proved by experiments conducted on various real-world datasets.

Nowadays, many recommendation systems are based on spatio-temporal information, such as POI recommendations. In the future work, we would like to apply adversarial transfer learning to POI recommender systems and combine adversarial transfer learning with spatio-temporal information to improve the recommendation performance.

References

[1] Srivastava R, Palshikar G K, Chaurasia S, Dixit A M. What's next? A recommendation system for industrial training. *Data Science and Engineering*, 2018, 3(3): 232-247.

[2] Zhao P, Zhu H, Liu Y, Xu J, Li Z, Zhuang F, Sheng V S, Zhou X. Where to go next: A spatio-temporal gated network for next POI recommendation. In *Proc. the 33rd AAAI Int. Conf. Artificial Intelligence*, January 2019, pp.5877-5884.

[3] Li B, Yang Q, Xue X. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In *Proc. the 21st Int. Joint Conf. Artificial Intelligence*, July 2009, pp.2052-2057.

[4] Cantador I, Fernández-Tobías I, Berkovsky S, Cremonesi P. Cross-domain recommender systems. In *Recommender Systems Handbook*, Ricci F, Rokach L, Shapira B (eds.), Springer, 2015, pp.919-959.

[5] Liu J, Zhao P, Zhuang F, Liu Y, Sheng V S, Xu J, Zhou X, Xiong H. Exploiting aesthetic preference in deep cross networks for cross-domain recommendation. In *Proc. the 29th World Wide Web Conference*, April 2020, pp.2768-2774.

[6] Berkovsky S, Kuik T, Ricci F. Mediation of user models for enhanced personalization in recommender systems. *User Model. User-Adapt. Interact.*, 2008, 18(3): 245-286.

[7] Fernández-Tobías I, Cantador I. Exploiting social tags in matrix factorization models for cross-domain collaborative filtering. In *Proc. the 1st Workshop on New Trends in Content-based Recommender Systems Co-Located with the 8th ACM Conf. Recommender Systems*, October 2014, pp.34-41.

[8] Tan S, Bu J, Qin X, Chen C, Cai D. Cross domain recommendation based on multi-type media fusion. *Neurocomputing*, 2014, 127: 124-134.

[9] Man T, Shen H, Jin X, Cheng X. Cross-domain recommendation: An embedding and mapping approach. In *Proc. the 26th Int. Joint Conf. Artificial Intelligence*, August 2017, pp.2464-2470.

[10] Zhu F, Wang Y, Chen C, Liu G, Orgun M A, Wu J. A deep framework for cross-domain and cross-system recommendations. In *Proc. the 27th Int. Joint Conf. Artificial Intelligence*, July 2018, pp.3711-3717.

[11] Yang Q, Hu G, Zhang Y. CoNet: Collaborative cross networks for cross-domain recommendation. In *Proc. the 27th ACM Int. Conf. Information and Knowledge Management*, October 2018, pp.667-676.

[12] Huang L, Zhao Z L, Wang C D, Huang D, Chao H Y. LSCD: Low-rank and sparse cross-domain recommendation. *Neurocomputing*, 2019, 366: 86-96.

[13] Loni B, Shi Y, Larson M A, Hanjalic A. Cross-domain collaborative filtering with factorization machines. In *Proc. the 36th European Conf. Information Retrieval Research*, April 2014, pp.656-661.

[14] Zhang F, Yuan N J, Lian D, Xie X, Ma W Y. Collaborative knowledge base embedding for recommender systems. In *Proc. the 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, August 2016, pp.353-362.

[15] Wang W, Yin H, Du X, Hua W, Li Y, Nguyen Q V H. Online user representation learning across heterogeneous social networks. In *Proc. the 42nd ACM SIGIR Int. Conf. Research and Development in Information Retrieval*, July 2019, pp.545-554.

[16] Ma J, Wen J, Zhong M, Chen W, Li X. MMM: Multi-source multi-net micro-video recommendation with clustered hidden item representation learning. *Data Science and Engineering*, 2019, 4(3): 240-253.

[17] Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. A survey on deep transfer learning. In *Proc. the 27th Int. Conf. Artificial Neural Networks*, October 2018, pp.270-279.

[18] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A C, Bengio Y. Generative adversarial nets. In *Proc. the 2014 Annual Conf. Neural Information Processing Systems*, December 2014, pp.2672-2680.

[19] Yuan F, Yao L, Benatallah B. DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. In *Proc. the 28th Int. Joint Conf. Artificial Intelligence*, August 2019, pp.4227-4233.

[20] Wang C, Niepert M, Li H. RecSys-DAN: Discriminative adversarial networks for cross-domain recommender systems. *IEEE Transactions on Neural Networks and Learning Systems*. doi:10.1109/TNNLS.2019.2907430.

[21] Kanagawa H, Kobayashi H, Shimizu N, Tagami Y, Suzuki T. Cross-domain recommendation via deep domain adaptation. In *Proc. the 41st European Conf. Information Retrieval*, April 2019, pp.20-29.

[22] He X, Liao L, Zhang H, Nie L, Hu X, Chua T S. Neural collaborative filtering. In *Proc. the 26th Int. Conf. World Wide Web*, April 2017, pp.173-182.

[23] Berkovsky S, Kuik T, Ricci F. Cross-domain mediation in collaborative filtering. In *Proc. the 11th Int. Conf. User Modeling*, June 2007, pp.355-359.

[24] Singh A P, Gordon G J. Relational learning via collective matrix factorization. In *Proc. the 14th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, August 2008, pp.650-658.

[25] Denton E L, Chintala S, Szlam A, Fergus R. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Proc. the 2015 Annual Conf. Neural Information Processing Systems*, December 2015, pp.1486-1494.

[26] Zhang Y, Barzilay R, Jaakkola T S. Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics*, 2017, 5: 515-528.

[27] Gui T, Zhang Q, Huang H, Peng M, Huang X. Part-of-speech tagging for twitter with adversarial neural networks. In *Proc. the 2017 Conf. Empirical Methods in Natural Language Processing*, September 2017, pp.2411-2420.

[28] Xu C, Zhao P, Liu Y, Sheng V S, Xu J, Zhuang F, Fang J, Zhou X. Graph contextualized self-attention network for session-based recommendation. In *Proc. the 28th Int. Joint Conf. Artificial Intelligence*, August 2019, pp.3940-3946.

[29] Zhang T, Zhao P, Liu Y, Sheng V S, Xu J, Wang D, Liu G, Zhou X. Feature-level deeper self-attention network for sequential recommendation. In *Proc. the 28th Int. Joint Conf. Artificial Intelligence*, August 2019, pp.4320-4326.

[30] Chen J, Zhang H, He X, Nie L, Liu W, Chua T S. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *Proc. the 40th ACM SIGIR Int. Conf. Research and Development in Information Retrieval*, August 2017, pp.335-344.

[31] Cheng Z, Ding Y, He X, Zhu L, Song X, Kankanhalli M S. A^3NCF: An adaptive aspect attention model for rating prediction. In *Proc. the 27th Int. Joint Conf. Artificial Intelligence*, July 2018, pp.3748-3754.

[32] Shi S, Zhang M, Liu Y, Ma S. Attention-based adaptive model to unify warm and cold starts recommendation. In *Proc. the 27th ACM Int. Conf. Information and Knowledge Management*, October 2018, pp.127-136.

[33] Xi W D, Huang L, Wang C D, Zheng Y Y, Lai J. B-PAM: Recommendation based on BP neural network with attention mechanism. In *Proc. the 28th Int. Joint Conf. Artificial Intelligence*, August 2019, pp.3905-3911.

[34] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In *Proc. the 8th IEEE Int. Conf. Data Mining*, December 2008, pp.263-272.

[35] Pan R, Zhou Y, Cao B, Liu N N, Lukose R M, Scholz M, Yang Q. One-class collaborative filtering. In *Proc. the 8th IEEE Int. Conf. Data Mining*, December 2008, pp.502-511.

[36] Ganin Y, Lempitsky V S. Unsupervised domain adaptation by backpropagation. In *Proc. the 32nd Int. Conf. Machine Learning*, July 2015, pp.1180-1189.

[37] He R, McAuley J J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proc. the 25th Int. Conf. World Wide Web*, April 2016, pp.507-517.

[38] Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. the 25th Conf. Uncertainty in Artificial Intelligence*, June 2009, pp.452-461.

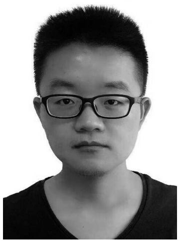
[39] Misra I, Shrivastava A, Gupta A, Hebert M. Cross-stitch networks for multi-task learning. In *Proc. the 2016 IEEE Conf. Computer Vision and Pattern Recognition*, June 2016, pp.3994-4003.

Ying Li received her B.S. degree in computer science from Soochow University, Suzhou, in 2018. She is currently a Master student in the School of Computer Science and Technology at Soochow University, Suzhou. Her research interests include data mining and recommender systems.

Jia-Jie Xu is an associate professor at the School of Computer Science and Technology, Soochow University, Suzhou. He got his Ph.D. and M.S. degrees in computer science from the Swinburne University of Technology, Victoria, and the University of Queensland, Brisbane, in 2011 and 2006 respectively. Before joining Soochow University in 2013, he worked as an assistant professor in Institute of Software, Chinese Academy of Sciences, Beijing. His research interests mainly include spatiotemporal database systems, big data analytics, mobile computing, and workflow systems.

Peng-Peng Zhao is an associate professor in the School of Computer Science and Technology at Soochow University, Suzhou. He received his Ph.D. degree in computer science from Soochow University, Suzhou, in 2008. His current research interests are in the study of data mining, deep learning, recommender systems, and big data analytics.

Jun-Hua Fang is a lecturer in the School of Computer Science and Technology, Soochow University, Suzhou, working with Prof. Xiaofang Zhou since August 2017. He received his Ph.D. degree in computer science from East China Normal University, Shanghai, in 2017. His research interests include distributed stream processing, cloud computing, and spatiotemporal database.



Wei Chen is currently a lecturer in the School of Computer Science and Technology, Soochow University, Suzhou. He received his Ph.D. degree in computer science from Soochow University, Suzhou, in 2018. His research interests include heterogeneous information network analysis, cross-platform linkage and recommendation, spatiotemporal database, knowledge graph embedding and refinement.

Lei Zhao is a professor in the School of Computer Science and Technology at Soochow University, Suzhou. He received his Ph.D. degree in computer science from Soochow University, Suzhou, in 2006. His research focuses on graph databases, social media analysis, query outsourcing, parallel and distributed computing. His recent research is to analyze large graph database in an effective, efficient, and secure way.