



Title	Semantic Relation Transfer for Non-overlapped Cross-domain Recommendations
Author(s)	Li, Zhi; Amagata, Daichi; Zhang, Yihong et al.
Citation	Lecture Notes in Computer Science. 2023, 13937, p. 271-283
Version Type	AM
URL	https://hdl.handle.net/11094/105170
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

Semantic Relation Transfer for Non-overlapped Cross-domain Recommendations

Zhi Li¹, Daichi Amagata¹, Yihong Zhang¹, Takahiro Hara¹, Shuichiro Haruta²,
Kei Yonekawa², and Mori Kurokawa²

¹ Osaka University, Japan

{li.zhi, amagata.daichi, hang.yihong, hara}@ist.osaka-u.ac.jp

² KDDI Research, Inc., Japan

{sh-haruta, ke-yonekawa, mo-kurokawa}@kddi-research.jp

Abstract. Although cross-domain recommender systems (CDRSs) are promising approaches to solving the cold-start problem, most CDRSs require overlapped users, which significantly limits their applications. To remove the overlap limitation, researchers introduced domain adversarial learning and embedding attribution alignment to develop non-overlapped CDRSs. Existing non-overlapped CDRSs, however, have several drawbacks. They ignore the semantic relations between source and target items, leading to noisy knowledge transfer. Moreover, they learn knowledge from both domain-shared and domain-specific preferences and are hence easily misled by the source-domain-specific preferences. To overcome these drawbacks, we propose a novel semantic relation-based knowledge transfer framework (SRTrans). We semantically cluster the source and the target items and calculate their similarities to extract relational knowledge between domains. To transfer the relational knowledge, we develop a new two-tier graph transfer network. Last, we introduce a task-oriented knowledge distillation supervision and combine it with a prediction loss to alleviate the negative impact of the source-domain-specific preferences. Our experimental results on real-world datasets demonstrate that SRTrans significantly outperforms state-of-the-art models.

Keywords: non-overlapped cross-domain recommendation · graph neural network · knowledge distillation.

1 Introduction

The cold-start problem is a general challenge in practical recommender systems [8, 14, 15]. Cross-domain recommendation (CDR) is a promising solution to alleviate the cold-start problem, because it utilizes sufficient data from a source domain as prior knowledge to support the recommendation in the sparse target domain [7, 10, 17]. Most cross-domain recommender systems (CDRSs) require overlapped users to transfer individual-level knowledge across domains. These methods consider different categories of items in one service (e.g., books and movies on Amazon’s e-commerce platform) as different domains [17]. In this scenario, overlapped users are easily collected and can serve as a bridge to connect

different domains and also to transfer knowledge. However, matching users in different services is infeasible, particularly when services are provided by different companies. The aforementioned CDRSs cannot make use of source data from external services because of a lack of overlapped users.

To make use of data from external services, non-overlapped CDRSs were developed. These systems transfer statistic-level knowledge and hence can remove the requirement for overlapped users. Domain adversarial learning (DAL) and embedding attribution alignment (EAA) are two mainstream approaches to constructing these CDRSs. DAL basically introduces a domain discriminator to extract domain-independent user and item embeddings [13]. By doing so, users' preferences are better modeled by utilizing both source and target interactions. EAA aligns the attribution of user and item embeddings between source and target domains to guide the target embedding learning with the source knowledge [1, 6, 10]. However, the state-of-the-art DAL and EAA methods [1, 10, 13] ignore semantic relations between source and target items and transfer knowledge from *all* source data. This introduces noisy source knowledge and results in a negative transfer and performance degradation, because user preferences contain both domain-shared and domain-*specific* parts [9].

Motivated by the above observations, we propose a novel semantic relation-based knowledge transfer framework (SRTrans) for non-overlapped CDRSs. Different from the state-of-the-art DAL and EAA methods, which extract and transfer knowledge from *all* source data including irrelevant noise, SRTrans introduces a new two-tier graph transfer network to transfer only the relational knowledge from the source to target domains (i.e., to alleviate the negative impact of domain-specific preferences). SRTrans combines a prediction loss with a new task-oriented knowledge distillation supervision. This distills the domain-shared preferences from the knowledge learned by the two-tier graph transfer network. By transferring only the source-domain-*shared* preferences, SRTrans can mitigate the negative impact of the source-domain-specific preferences and can transfer knowledge better than the state-of-the-art methods. In summary, our contributions are three-fold:

- 1 We propose a semantic relation-based graph transfer framework, namely SRTrans, for non-overlapped CDRSs. SRTrans is robust against noisy source data and can extract useful source knowledge.
- 2 To mitigate the negative impact of source-domain-specific preferences, we introduce a new task-oriented knowledge distillation supervision and combine it with a prediction loss. This approach improves the final performance of SRTrans.
- 3 We conduct extensive experiments on real-world datasets, and the results demonstrate that SRTrans outperforms the state-of-the-art methods.

2 Related Work

User-overlapped CDR. Most CDRSs bridge source and target domains through overlapped users. These systems transfer individual-level knowledge learned in

the source domain to the target domain. For example, BiTGCF [9] and GADTCDR [16] respectively introduced a graph neural network and an attention mechanism as a cross-domain feature transfer layer to fuse overlapped users' source and target latent features. However, the assumption of overlapped users limits their applications.

Another research direction is to learn mappings between source and target user embeddings. EMCDR [11] proposed a multi-layer perceptron to map source user embeddings to target embeddings. PTUPCDR [17] further improved EMCDR by learning personalized meta-transfer mappings.

Non-overlapped CDR. To remove the limitation of overlapped users, non-overlapped CDRSs were developed, and domain adversarial learning (DAL) for non-overlapped CDRSs was introduced. For example, RecSys-Dan [13] devised a discriminator and minimized the divergence of the predictions between source and target domains. Some studies proposed embedding attribution alignment (EAA), which aligns embedding attributions between source and target domains. MMT-Net [6] developed a contextual CDRS and regularized the contextual-jointed target user and item embedding learning with learned source embedding distributions. ESAM [1] and CFAA [10] removed the requirement for domain-shared contextual features and aligned the attribution distribution and correlation between source and target domains.

However, existing DAL and EAA approaches ignore the relations between source and target and leverage all source data to transfer knowledge, resulting in an impaired transfer or even a negative transfer. These approaches, moreover, learn from both domain-shared and domain-specific preferences and yield sub-optimal performances. The above drawbacks motivate us to propose SRTrans, which extracts relational knowledge and alleviates the negative impact of the domain-specific preferences in the source domain.

3 Method

Problem Formulation. We first formulate the cross-domain recommendation task as the top-K recommendation in a sparse target domain $\mathcal{D}_t$, under the assumption of the existence of an auxiliary dense domain $\mathcal{D}_s$ that is considered to be the source. Let $\mathcal{U}_t$ and $\mathcal{V}_t$ denote sets of users and items in $\mathcal{D}_t$, respectively. The interaction set between $\mathcal{U}_t$ and $\mathcal{V}_t$ is denoted as $\mathcal{R}_t = \{(u, v) | u \in \mathcal{U}_t, v \in \mathcal{V}_t\}$. Analogously, we denote the user set, item set, and interaction set in $\mathcal{D}_s$ as $\mathcal{U}_s$, $\mathcal{V}_s$, and $\mathcal{R}_s$, respectively. It is important to note that there is no overlap of user and item between $\mathcal{D}_s$ and $\mathcal{D}_t$. Given a user $u \in \mathcal{U}_t$, the top-K recommendations aim to predict a preference score $\hat{y}_{u,v}$ for each item $v \in \bar{\mathcal{V}}_u = \{v | v \in \mathcal{V}_t, v \notin \mathcal{V}_u\}$, where $\mathcal{V}_u$ is the set of items that interacted with u . The preference score $\hat{y}_{u,v}$ is defined as $\hat{y}_{u,v} = f(\mathbf{e}_u, \mathbf{e}_v)$, where $f(\cdot)$ is a user preference estimator, $\mathbf{e}_u$ is the user embedding, and $\mathbf{e}_v$ is the item embedding. The K items with the largest scores are recommended to u .

Overview. Motivated by the previous non-overlapped CDRSs that ignore the relations among interactions and are easily misled by the source-domain-specific

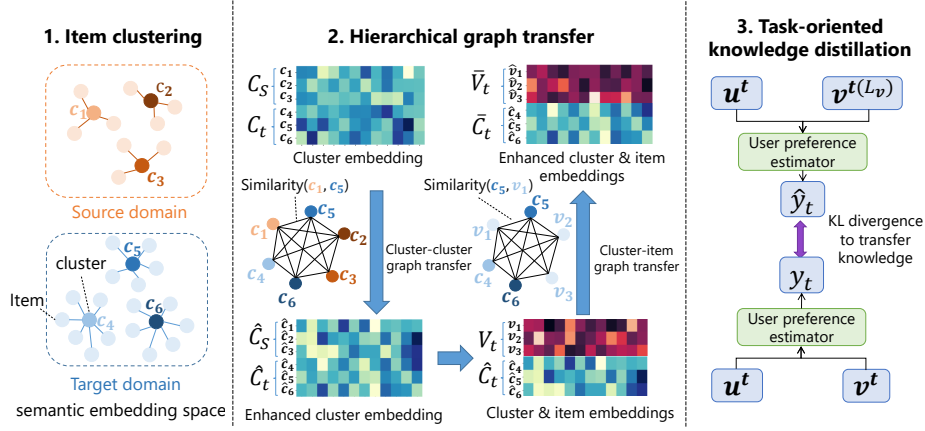


Fig. 1. Overview of our proposed SRTrans

preferences, we propose a novel semantic relation-based graph transfer framework (SRTrans) that extracts and transfers cluster-based relational knowledge while alleviating the misleading source-domain-specific preferences. SRTrans is depicted in Fig. 1. The numbers below correspond to the ones in Fig. 1.

(1) We first semantically cluster items and calculate cluster embeddings through an adaptive cluster approach. (2) Then, we calculate similarities of these clusters to measure their relations. We also propose a novel two-tier graph transfer network that extracts and transfers relational knowledge based on the cluster relations, where the relational knowledge is finally aggregated into target items by graph neural networks. (3) After that, we design a new task-oriented knowledge distillation supervision that can be combined with a prediction loss to avoid learning the source-domain-specific preferences.

3.1 Adaptive Semantic Item Cluster

To learn semantic item clusters, we first compute the semantic embeddings of items. Given the item texts, i.e., item descriptions, we apply the semantic token embeddings from a pre-trained BERT model [2] to represent the tokens in the item texts. Let T_v denote the text of an item v , and the semantic embedding of v is the *tf-idf* weighted aggregation of token embeddings:

$$\mathbf{v}_{txt} = \sum_{w \in T_v} tf-idf(w) \cdot \text{BERT}(w), \quad (1)$$

where $\text{BERT}(w)$ and $tf-idf(w)$ are respectively the embedding and the *tf-idf* score for token $w \in T_v$. The *tf-idf* score is calculated with combination of the source and target item text corpora. By doing so, we can focus on important tokens and alleviate the negative impact of noisy tokens. The dimension of the BERT semantic embeddings (768) is so high and may lead to tremendous computing costs and over-fitting issues in our sparse target domain. We hence assign

a dense layer ϕ_{emb} with output dimension $d \ll 768$ to encode $\mathbf{v}_{txt}$. The encoded item embedding $\mathbf{v}$ is given by

$$\mathbf{v} = \phi_{emb}(\mathbf{v}_{txt}). \quad (2)$$

After that, we calculate semantic embeddings for all source and target items, which are denoted as $\mathbf{V}^s$ and $\mathbf{V}^t$, respectively.

To get semantic item clusters that can facilitate the final recommendation task, we borrow the idea of DE-RRD [4]. Formally, given the cluster number $\mathcal{N}_c$, we calculate the cluster assignment probability vector $\mathbf{p} \in \mathbb{R}^{\mathcal{N}_c}$ with a small network ϕ_{ca} , where the vector of an item v is given by

$$\mathbf{p}_v = \phi_{ca}(\mathbf{v}). \quad (3)$$

Each element $p_{v,j}$ of $\mathbf{p}_v$ represents the probability that the item v is assigned to the cluster j . With the help of $\mathbf{p}_v$, we assign a binary gradient vector $\mathbf{m}_v \in \mathbb{R}^{\mathcal{N}_c}$ to indicate the cluster of item v , where the element j of $\mathbf{m}_v$ is given by

$$m_{v,j} \sim \text{Bern}\left(\frac{\exp(p_{v,j})}{\sum_{k \in \mathcal{N}_c} \exp(p_{v,k})}\right), \quad (4)$$

where $\text{Bern}(\cdot)$ represents the Bernoulli distribution. We use the Gumbel-Softmax reparameterization trick to differentiate through the Bernoulli sampling process:

$$m_{v,j} = \frac{\exp((p_{v,j}) + g_j/\tau)}{\sum_{k \in \mathcal{N}_c} \exp((p_{v,k}) + g_k/\tau)}, \quad \mathbf{g} \sim \text{Gumbel}(0, \mathbf{1}) \quad (5)$$

where $\mathbf{g} \in \mathbb{R}^{\mathcal{N}_c}$ is the Gumbel noise drawn from $\text{Gumbel}(0, \mathbf{1})$ distribution. Note that g_j is the j -th element of $\mathbf{g}$ and τ is the temperature parameter. The sampling process is separated from $\mathbf{p}_v$, so the cluster assigning network ϕ_{ca} can be end-to-end updated through backpropagations. Let $\mathbf{M}^s \in \mathbb{R}^{|\mathcal{V}^s| \times \mathcal{N}_c}$ ($\mathbf{M}^t \in \mathbb{R}^{|\mathcal{V}^t| \times \mathcal{N}_c}$) denote the binary gradient matrix for the source (target) domain, where each row in $\mathbf{M}^s$ ($\mathbf{M}^t$) indicates the cluster of a source (target) item and can be computed by Equations (3) and (5).

We next declare source cluster embeddings $\mathbf{C}^s$ and target cluster embeddings $\mathbf{C}^t$ as the average semantic embeddings of the items assigned to the corresponding cluster:

$$\mathbf{C}^s = (\tilde{\mathbf{M}}^s)^T \mathbf{V}^s \quad \text{and} \quad \mathbf{C}^t = (\tilde{\mathbf{M}}^t)^T \mathbf{V}^t, \quad (6)$$

where $\tilde{\mathbf{M}}^s$ and $\tilde{\mathbf{M}}^t$ respectively are assignment matrices normalized by the number of items in each cluster, i.e., $\tilde{\mathbf{M}}_{[:,i]}^s = \mathbf{M}_{[:,i]}^s / \sum_i \mathbf{M}_{[:,i]}^s$.

3.2 Two-tier Graph Transfer

We propose a novel two-tier graph transfer network to extract and transfer cluster-based relational knowledge from the source to the target. This network consists of a cluster-level relational graph transfer and a cluster-item relational graph transfer. The former transfers knowledge from the source to the target

clusters by constructing a relational graph whose adjacency matrix is denoted as the similarities between the source and target clusters. Analogously, the latter transfers knowledge from the target clusters to the target items by building a graph whose adjacency matrix is denoted as the similarities between the target clusters and items. With these similarity-based adjacency matrices, the graph transfer network can transfer the most relevant source knowledge from source clusters into the target items and thus alleviate the negative impact of irrelevant noises. Therefore, we can remove the drawbacks of the state-of-the-art methods (transferring all source knowledge, including domain-specific knowledge).

Cluster-level Relational Graph Transfer. To extract and transfer cluster-level relational knowledge from the source to the target, we first define a cluster-based relational graph $\mathcal{G}_c$ to represent the relations between clusters from different domains. Let $\mathbf{C} = (\mathbf{C}^s; \mathbf{C}^t) \in \mathbb{R}^{(2\mathcal{N}_c) \times d}$ denote the fused source and target cluster representations and serve as vertices in graph $\mathcal{G}_c$. We further define the edges and the corresponding edge weights. The edge weight $\mathcal{S}(\mathbf{c}_i, \mathbf{c}_j)$ between clusters $\mathbf{c}_i$ and $\mathbf{c}_j$ is measured by the cosine similarity with softmax and is formulated by:

$$\mathcal{S}(\mathbf{c}_i, \mathbf{c}_j) = \text{softmax}\left(\frac{\mathbf{c}_i \cdot \mathbf{c}_j}{\|\mathbf{c}_i\| \|\mathbf{c}_j\|}\right). \quad (7)$$

For simplicity, we denote the cluster-based relational graph as $\mathcal{G}_c = (\mathbf{C}, \mathcal{A})$, where $\mathbf{C} = \{\mathbf{c}_i | i \in [1, 2\mathcal{N}_c]\} \in \mathbb{R}^{2\mathcal{N}_c \times d}$ represents the set of vertices and each vertex corresponds to a cluster, whereas $\mathcal{A} = \{\mathcal{S}(\mathbf{c}_i, \mathbf{c}_j) | \mathbf{c}_i \in \mathbf{C}, \mathbf{c}_j \in \mathbf{C}\}$ is the adjacency matrix, which indicates the relations between clusters. Note that $\mathcal{A} = \{\mathcal{S}(\mathbf{C}^s, \mathbf{C}^s), \mathcal{S}(\mathbf{C}^s, \mathbf{C}^t); \mathcal{S}^T(\mathbf{C}^s, \mathbf{C}^t), \mathcal{S}(\mathbf{C}^t, \mathbf{C}^t)\}$ contains cluster relations within and across domains.

After constructing the cluster-based relational graph $\mathcal{G}_c$, we use the graph convolutional network (GCN) [5] to transfer relational knowledge from source clusters $\mathbf{C}^s$ to target clusters $\mathbf{C}^t$, where the GCN is formulated as:

$$\mathbf{C}^{(l+1)} = \text{ReLU}\left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{C}^{(l)} \mathbf{W}_c^{(l)}\right). \quad (8)$$

Here $\tilde{\mathcal{A}} = \mathcal{A} + \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, corresponds to adding self loops to the graph. Also, $\tilde{\mathbf{D}}$ is the degree matrix with elements $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and $\mathbf{W}_c^{(l)}$ is a trainable weight matrix for layer l . The input $\mathbf{C}^{(0)} = \mathbf{C}$. After aggregating L_c GCN layers, we get the relational enhanced target cluster representations as the last $\mathcal{N}_c$ rows of $\mathbf{C}^{(L_c)}$, which is denoted as $\mathbf{C}^{t(L_c)} = \{\mathbf{c}_i^{t(L_c)} | i \in [1, \mathcal{N}_c]\}$. By fusing and aggregating source and target clusters together, the enhanced target clusters can refine relational knowledge within and across domains.

Cluster-item Relational Graph Transfer. After calculating the enhanced target clusters $\mathbf{C}^{t(L_c)}$, we further construct a cluster-item relational graph $\mathcal{G}_v$ to measure the relations between enhanced target clusters $\mathbf{C}^{t(L_c)}$ and the target items $\mathbf{V}^t$. With $\mathcal{G}_v$ and cluster-item relations, the useful knowledge from the most relevant $\mathbf{C}^{t(L_c)}$ is aggregated into the target items, while the irrelevant noise is alleviated. To fuse knowledge from both target clusters and items, we

concatenate $\mathbf{C}^{t(L_c)}$ and $\mathbf{V}^t$ to serve as vertices in the graph $\mathcal{G}_v$, where the vertex set is denoted as $\mathbf{H} = (\mathbf{C}^{t(L_c)}; \mathbf{V}^t) \in \mathbb{R}^{(\mathcal{N}_c + |\mathcal{V}_t|) \times d}$. Analogously, the edge weight $\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j)$ between vertex $\mathbf{h}_i$ and $\mathbf{h}_j$ is defined as the cosine similarity with softmax:

$$\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) = \text{softmax}\left(\frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \|\mathbf{h}_j\|}\right). \quad (9)$$

The cluster-item relational graph $\mathcal{G}_v$ is denoted as $\mathcal{G}_v = (\mathbf{H}, \mathcal{B})$, where the vertex set and the adjacency matrix are $\mathbf{H} = \{\mathbf{h}_i | i \in [1, \mathcal{N}_c + |\mathcal{V}_t|]\}$ and $\mathcal{B} = \{\mathcal{S}(\mathbf{h}_i, \mathbf{h}_j) | \mathbf{h}_i \in \mathbf{H}, \mathbf{h}_j \in \mathbf{H}\}$, respectively.

By leveraging $\mathcal{G}_v$, we next transfer relational knowledge via a GCN, which is formulated as:

$$\mathbf{H}^{(l+1)} = \text{ReLU}\left(\tilde{\mathbf{D}}_v^{-\frac{1}{2}} \tilde{\mathcal{B}} \tilde{\mathbf{D}}_v^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}_v^{(l)}\right). \quad (10)$$

Here, $\tilde{\mathcal{B}}$ is the adjacency matrix with self loops, and $\tilde{\mathbf{D}}_v$ is the degree matrix on $\tilde{\mathcal{B}}$. $\mathbf{W}_v^{(l)}$ is a trainable weight matrix for layer l . Also, we define $\mathbf{H}^{(0)} = \mathbf{H}$. After aggregating L_v GCN layers, we get the knowledge-enhanced target item embeddings $\mathbf{V}^{t(L_v)}$ as the last $|\mathcal{V}_t|$ rows of $\mathbf{W}_v^{(L_v)}$.

3.3 Task-oriented Knowledge Distillation

Existing non-overlapped CDRSs [1, 13, 10] are easily misled by domain-specific preferences, because the source and target embedding space is directly aligned. Motivated by this finding, we combine a knowledge distillation with the target prediction by introducing a new task-oriented knowledge distillation supervision. In this way, the target prediction loss can supervise the knowledge distillation to learn the domain-*share* preferences. As a result, the negative transfer issues incurred by the misleading source-domain-specific preferences can be alleviated.

Formally, we first define the user embeddings of source and target domains as trainable parameter matrices $\mathbf{U}^s$ and $\mathbf{U}^t$, respectively. Each row in $\mathbf{U}^s$ ($\mathbf{U}^t$) corresponds to a source (target) user and can be retrieved with the user ID. The prediction score of the target user u^t for the target item v^t before and after knowledge enhancement in Section 3.2 are then given by:

$$\hat{y}_{u,v}^t = f(\mathbf{u}^t, \mathbf{v}^t) \quad \text{and} \quad \tilde{y}_{u,v}^t = f(\mathbf{u}^t, \mathbf{v}^{t(L_v)}), \quad (11)$$

respectively, where $\mathbf{v}^t$ and $\mathbf{v}^{t(L_v)}$ are respectively the item embedding before and after knowledge enhancement. For a user preference estimator $f(\cdot)$, we adopt inner product [12] and LGC [3] to evaluate the performance of our SRTrans on different models. After that, the task-oriented knowledge distillation is defined as the KL-divergence between $\hat{y}_{u,v}^t$ and $\tilde{y}_{u,v}^t$, which is formulated by:

$$\mathcal{L}_{KD}(\hat{y}^t | \tilde{y}^t) = \sum_{(u,v) \in \mathcal{R}^t} \left(\hat{y}_{u,v}^t \log \frac{\hat{y}_{u,v}^t}{\tilde{y}_{u,v}^t} + (1 - \hat{y}_{u,v}^t) \log \frac{(1 - \hat{y}_{u,v}^t)}{(1 - \tilde{y}_{u,v}^t)} \right). \quad (12)$$

We use the pair-wise BPR loss [3] to measure the loss of predictions. To achieve this, we randomly sample a negative item for each source and each target interaction. Taking the target domain as an example, a new interaction $r^t \in \mathcal{R}_t$ is

a triplet $r^t = (u, v, v')$, where $u \in \mathcal{U}_t$, $v \in \mathcal{V}_t$, and $v' \in \bar{\mathcal{V}}_u$. Then, the pair-wise BPR loss is given by

$$\mathcal{L}_{BPR} = - \left(\sum_{(u,v,v') \in \mathcal{R}_t} \ln \sigma(\hat{y}_{uv} - \hat{y}_{uv'}) + \sum_{(u,v,v') \in \mathcal{R}_s} \ln \sigma(\hat{y}_{uv} - \hat{y}_{uv'}) \right). \quad (13)$$

The total loss is measured by combining the knowledge distillation loss $\mathcal{L}_{KD}$ and the prediction loss $\mathcal{L}_{BPR}$, that is

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda \mathcal{L}_{KD}, \quad (14)$$

where λ is a hyper-parameter used to balance the weights of different losses. By combining $\mathcal{L}_{KD}$ and $\mathcal{L}_{BPR}$, the negative impact from the source-domain-specific preferences can be alleviated under the supervision of the prediction loss.

4 Experiment

4.1 Experiment Setting

Datasets. We used two public and two private datasets to investigate the recommendation performance of SRTrans in practical applications and for benchmarking purposes.

The public datasets were MovieLens25M³ and Amazon⁴. For MovieLens25M (ML), we used the movie ratings from *30/9/2016* to *1/10/2018*, where the movie descriptions were collected from TMDb⁵. For Amazon, we built the Amazon-Book (AB) subset containing ratings for books from *30/9/2016* to *3/10/2018*.

The private datasets have an online advertisement dataset (AD) and an e-commerce dataset (E-com). AD contains Web browsing records from *1/8/2017* to *31/8/2017* on an ads platform and the textual content of Web pages. E-com has the purchase records on an e-commerce platform and the textual descriptions of products; purchase records in E-com have the same period as that of AD.

Each CDR scenario consisted of a source domain and a relatively sparse target domain. The CDR scenario from domain A to domain B is denoted as $A \rightarrow B$, where A and B are the source domain and the target domain, respectively. All the CDR scenarios included: (1) $AD \rightarrow E\text{-com}$ and $E\text{-com} \rightarrow AD$; (2) $ML \rightarrow AB$ and $AB \rightarrow ML$. For each source domain, we selected users with 3 to 10 interaction records and items with 10 to 15 interaction records to adapt to a dense setting. Inversely, for each target domain, we selected users with 3 to 5 interactions and items with 5 to 15 interactions to form a relatively sparse environment. Some basic information on the pre-processed datasets is summarized in Table 1.

Evaluation Criteria. For each user in the target domains, we took her last and second-last interactions to form the test and validation sets, respectively. The

³ grouplens.org/datasets/movielens/25m/

⁴ jmcauley.ucsd.edu/data/amazon/

⁵ www.themoviedb.org/documentation/api

Table 1. Basic information on the datasets we used

	Dataset	#users	#items	#interactions	avg. #interactions per user
As source	ML	18,232	14,435	421,803	23.14
	AB	27,662	12,708	129,899	4.70
	AD	18,829	12,253	360,880	19.17
	E-com	17,418	6,142	81,499	4.68
As target	ML	6,298	9,873	31,445	4.99
	AB	13,350	10,477	61,004	4.57
	AD	11,010	12,031	55,050	5.00
	E-com	12,558	5,118	46,871	3.73

remaining interactions were used as the training set. Then, we ranked her test item with 99 randomly sampled negative items as the candidate recommendations, where negative items are items that have no interaction with this user. We employed widely used *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG) to measure the accuracy of the top-K recommendation.

Baselines. We compared *SRTrans*⁶ with the single-domain baseline (*BASE*) and the following state-of-the-art non-overlapped CDRSs: *RecSys-DAN* [13], *ESAM* [1], and *CFAA* [10]. Because these baselines and our *SRTrans* are model-agnostic frameworks, we adopted two base models for them: *BPR-MF* [12] (an MF-based model) and *LightGCN* [3] (a GNN-based model). For fair comparison, the embedding module of the baselines was replaced with our semantic encoded item embedding that can be calculated by Equation (2).

Implementation Details. We adopted the same L2 penalty and mini-batch trick for the evaluated methods and set them to 0.01 and 2048, respectively. The GCN layer for the base model *LightGCN* was set to 3. The embedding dimension d was 32 and 16 for public and private datasets, respectively. For *SRTrans*, the temperature parameter τ was 0.0001 and the weight of the knowledge distillation loss λ was 0.1. The number of graph transfer layers L_c and L_v was 1. The number of clusters $\mathcal{N}_c$ was 32. We implemented MF- and *LightGCN*-based *SRTrans*, *RecSys-DAN*, *ESAM*, and *CFAA* with PyTorch framework, using the Adam optimizer where the learning rate was set to 0.01. The above hyper-parameters were fine-tuned according to the performance on the validation set.

4.2 Experiment Results

Comparison. Table 2 shows the comparison results. We find that: (1) *SRTrans* outperforms the baselines w.r.t. HR@5 and NDCG@5 in most cases, especially when the base model is *LightGCN*. This is because *SRTrans* transfers relational knowledge into the individual item, and GNN-based models can further fuse knowledge in these items by structurally aggregating them. (2) Although cross-domain baselines transfer user preferences (*RecSys-DAN*) or align embeddings spaces (*ESAM* and *CFAA*) from the source domain to the target domain, they often perform worse than the single-domain method (*BASE*). This result indicates that domain-specific preferences and noisy source data incur negative

⁶ <https://github.com/ZL6298/SRTrans/>

Table 2. Comparison between our proposal and state-of-the-art. Performances $\pm 95\%$ confidence intervals are reported. Bold shows the winner.

	Method	AD $\rightarrow$ E-com		E-com $\rightarrow$ AD	
		HR@1	NDCG@5	HR@1	NDCG@5
MF	BASE	0.224 $\pm$ 0.033	0.310 $\pm$ 0.014	0.180 $\pm$ 0.008	0.299 $\pm$ 0.006
	RecSys-DAN	0.269 $\pm$ 0.025	0.356 $\pm$ 0.015	0.046 $\pm$ 0.010	0.102 $\pm$ 0.014
	ESAM	0.283$\pm$0.035	0.365$\pm$0.016	0.125 $\pm$ 0.012	0.239 $\pm$ 0.013
	CFAA	0.251 $\pm$ 0.036	0.336 $\pm$ 0.014	0.140 $\pm$ 0.008	0.257 $\pm$ 0.007
	SRTrans (ours)	0.243 $\pm$ 0.028	0.331 $\pm$ 0.011	0.187$\pm$0.005	0.307$\pm$0.004
LightGCN	BASE	0.297 $\pm$ 0.012	0.372 $\pm$ 0.009	0.190 $\pm$ 0.008	0.315$\pm$0.008
	RecSys-DAN	0.254 $\pm$ 0.016	0.277 $\pm$ 0.016	0.043 $\pm$ 0.006	0.065 $\pm$ 0.007
	ESAM	0.216 $\pm$ 0.033	0.314 $\pm$ 0.023	0.114 $\pm$ 0.010	0.221 $\pm$ 0.012
	CFAA	0.282 $\pm$ 0.035	0.375 $\pm$ 0.021	0.042 $\pm$ 0.004	0.114 $\pm$ 0.007
	SRTrans (ours)	0.312$\pm$0.011	0.384$\pm$0.009	0.191$\pm$0.010	0.306 $\pm$ 0.009
	Method	ML $\rightarrow$ AB		AB $\rightarrow$ ML	
		HR@1	NDCG@5	HR@1	NDCG@5
MF	BASE	0.072 $\pm$ 0.003	0.136 $\pm$ 0.004	0.031 $\pm$ 0.002	0.076 $\pm$ 0.002
	RecSys-DAN	0.021 $\pm$ 0.006	0.048 $\pm$ 0.007	0.005 $\pm$ 0.001	0.019 $\pm$ 0.002
	ESAM	0.034 $\pm$ 0.006	0.079 $\pm$ 0.008	0.032$\pm$0.003	0.083$\pm$0.004
	CFAA	0.055 $\pm$ 0.004	0.109 $\pm$ 0.006	0.029 $\pm$ 0.003	0.076 $\pm$ 0.004
	SRTrans (ours)	0.075$\pm$0.005	0.142$\pm$0.005	0.030 $\pm$ 0.002	0.075 $\pm$ 0.003
LightGCN	BASE	0.122 $\pm$ 0.008	0.194 $\pm$ 0.006	0.085 $\pm$ 0.007	0.165 $\pm$ 0.008
	RecSys-DAN	0.042 $\pm$ 0.009	0.084 $\pm$ 0.009	0.013 $\pm$ 0.001	0.041 $\pm$ 0.004
	ESAM	0.046 $\pm$ 0.010	0.119 $\pm$ 0.011	0.070 $\pm$ 0.011	0.166 $\pm$ 0.018
	CFAA	0.049 $\pm$ 0.010	0.121 $\pm$ 0.012	0.016 $\pm$ 0.001	0.049 $\pm$ 0.002
	SRTrans (ours)	0.129$\pm$0.011	0.204$\pm$0.009	0.107$\pm$0.008	0.202$\pm$0.009

transfer issues. (3) SRTrans achieves comparable performance or outperforms the single-domain model (BASE) in most scenarios. This observation confirms that SRTrans remarkably alleviates the negative transfer issue.

Visualization. To better show the knowledge transfer process and explain the learned cluster relations, we visualize the cluster-based relational graph by sampling a mini-batch of interactions from the training data. The result of ML $\rightarrow$ AB is shown in Fig. 2.

Fig. 2(a) depicts a heatmap of an adjacency matrix that describes the relation between the source and target clusters. The colors indicate cosine similarities between clusters. From this figure, we can find that highly related source and target clusters (indicated by light colors) and irrelevant ones (indicated by dark colors) are identified. Fig. 2(b) gives a sub-graph constructed by using 10 most related source-target cluster pairs. This figure shows how the knowledge in source clusters is transferred to target clusters. For example, target cluster 23 receives more knowledge from its highly relevant source clusters 13, 30, 7, and 14, compared to other irrelevant clusters. This example highlights the mechanism of alleviating negative transfer issues.

Ablation Study. To study how each module of SRTrans contributes to the final performance, we compared SRTrans with its several variants, namely (1) w/o SI, which replaces semantic item embeddings with randomly initialized ones, (2) w/o AC, which removes adaptive cluster module and directly calculates similarities between single source and target items to build knowledge transfer graph, and (3) w/o KD, which is SRTrans without knowledge distillation and is equal to the single-domain BASE model. Table 3 reports the result.

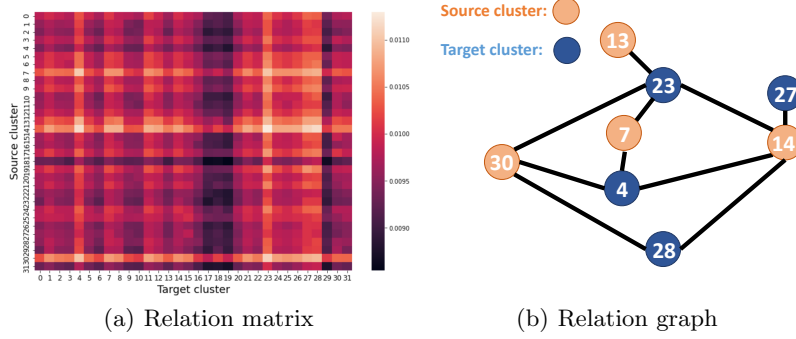


Fig. 2. Visualization result on a mini-batch of data from the ML $\rightarrow$ AB scenario

Table 3. Performances of variants of SRTrans

Method	AD $\rightarrow$ E-com		E-com $\rightarrow$ AD	
	HR@1	NDCG@5	HR@1	NDCG@5
w/o SI	0.292 $\pm$ 0.008	0.351 $\pm$ 0.006	0.090 $\pm$ 0.005	0.140 $\pm$ 0.005
w/o AC	0.310 $\pm$ 0.012	0.376 $\pm$ 0.009	0.194 $\pm$ 0.009	0.310 $\pm$ 0.008
w/o KD	0.297 $\pm$ 0.012	0.372 $\pm$ 0.009	0.190 $\pm$ 0.008	0.315 $\pm$ 0.008
SRTrans	0.312 $\pm$ 0.011	0.384 $\pm$ 0.009	0.191 $\pm$ 0.01	0.306 $\pm$ 0.009
Method	ML $\rightarrow$ AB		AB $\rightarrow$ ML	
	HR@1	NDCG@5	HR@1	NDCG@5
w/o SI	0.101 $\pm$ 0.008	0.159 $\pm$ 0.007	0.060 $\pm$ 0.004	0.115 $\pm$ 0.005
w/o AC	0.130 $\pm$ 0.011	0.207 $\pm$ 0.009	0.097 $\pm$ 0.007	0.188 $\pm$ 0.009
w/o KD	0.122 $\pm$ 0.008	0.194 $\pm$ 0.006	0.085 $\pm$ 0.007	0.165 $\pm$ 0.008
SRTrans	0.128 $\pm$ 0.011	0.201 $\pm$ 0.009	0.107 $\pm$ 0.008	0.202 $\pm$ 0.009

We see that w/o SI has the lowest performance. This result indicates that the semantic features are essential to extracting the relational knowledge. Moreover, w/o AC shows a comparable performance to SRTrans in some cases, suggesting that transferring item-based relational knowledge can also alleviate the performance degradation caused by source noise data.

5 Conclusion

This work proposed a novel semantic relation-based knowledge transfer framework (SRTrans) for non-overlapped cross-domain recommendations. SRTrans introduces a new two-tier graph transfer network that extracts relational knowledge from a source domain to enhance the target item embeddings. With these embeddings, SRTrans combines a task-oriented knowledge distillation loss with a prediction loss to adaptively learn from domain-shared preferences and to alleviate the negative impacts of source-domain-specific preferences. Our experimental results demonstrate the superiority of SRTrans.

Acknowledgement. This work partially supported by JST CREST Grant Number JPMJCR21F2.

References

1. Chen, Z., Xiao, R., Li, C., Ye, G., Sun, H., Deng, H.: ESAM: discriminative domain adaptation with non-displayed items to improve long-tail performance. In: SIGIR. pp. 579–588 (2020)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT. pp. 4171–4186 (2019)
3. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR. pp. 639–648 (2020)
4. Kang, S., Hwang, J., Kweon, W., Yu, H.: DE-RRD: A knowledge distillation framework for recommender system. In: CIKM. pp. 605–614 (2020)
5. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
6. Krishnan, A., Das, M., Bendre, M., Yang, H., Sundaram, H.: Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In: SIGIR. pp. 1081–1090 (2020)
7. Li, Z., Amagata, D., Zhang, Y., Hara, T., Haruta, S., Yonekawa, K., Kurokawa, M.: Debiasing graph transfer learning via item semantic clustering for cross-domain recommendations. In: IEEE Big Data. pp. 762–769 (2022)
8. Li, Z., Amagata, D., Zhang, Y., Maekawa, T., Hara, T., Yonekawa, K., Kurokawa, M.: Hml4rec: Hierarchical meta-learning for cold-start recommendation in flash sale e-commerce. *Knowledge-Based Systems* **255**, 109674 (2022)
9. Liu, M., Li, J., Li, G., Pan, P.: Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In: CIKM. pp. 885–894 (2020)
10. Liu, W., Zheng, X., Hu, M., Chen, C.: Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In: Web Conference. pp. 1181–1190 (2022)
11. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: An embedding and mapping approach. In: Sierra, C. (ed.) IJCAI. pp. 2464–2470 (2017)
12. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI. pp. 452–461 (2009)
13. Wang, C., Niepert, M., Li, H.: Recsys-dan: Discriminative adversarial networks for cross-domain recommender systems. *IEEE Trans. Neural Networks Learn. Syst.* **31**(8), 2731–2740 (2020)
14. Wang, H., Amagata, D., Maekawa, T., Hara, T., Niu, H., Yonekawa, K., Kurokawa, M.: Preliminary investigation of alleviating user cold-start problem in e-commerce with deep cross-domain recommender system. In: ECNLP. pp. 398–403 (2019)
15. Wang, H., Amagata, D., Maekawa, T., Hara, T., Hao, N., Yonekawa, K., Kurokawa, M.: A dnn-based cross-domain recommender system for alleviating cold-start problem in e-commerce. *IEEE Open Journal of the Industrial Electronics Society* **1**, 194–206 (2020)
16. Zhu, F., Wang, Y., Chen, C., Liu, G., Zheng, X.: A graphical and attentional framework for dual-target cross-domain recommendation. In: IJCAI. pp. 3001–3008 (2020)
17. Zhu, Y., Tang, Z., Liu, Y., Zhuang, F., Xie, R., Zhang, X., Lin, L., He, Q.: Personalized transfer of user preferences for cross-domain recommendation. In: WSDM. pp. 1507–1515 (2022)