Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users

SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu
Dept. of Computer Science and Engineering, POSTECH, South Korea
{seongku, jyhwang, dongha.lee, hwanjoyu}@postech.ac.kr

ABSTRACT

Providing accurate recommendations to newly joined users (or potential users, so-called *cold-start users*) has remained a challenging yet important problem in recommender systems. To infer the preferences of such cold-start users based on their preferences observed in other domains, several cross-domain recommendation (CDR) methods have been studied. The state-of-the-art Embedding and Mapping approach for CDR (EMCDR) aims to infer the latent vectors of cold-start users by supervised mapping from the latent space of another domain. In this paper, we propose a novel CDR framework based on semi-supervised mapping, called SSCDR, which effectively learns the cross-domain relationship even in the case that only a few number of labeled data is available. To this end, it first learns the latent vectors of users and items for each domain so that their interactions are represented by the distances, then trains a cross-domain mapping function to encode such distance information by exploiting both overlapping users as labeled data and all the items as unlabeled data. In addition, SSCDR adopts an effective inference technique that predicts the latent vectors of cold-start users by aggregating their neighborhood information. Our extensive experiments on different CDR scenarios show that SSCDR outperforms the state-of-the-art methods in terms of CDR accuracy, particularly in the realistic settings that a small portion of users overlap between two domains.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; • **Computing methodologies** → *Learning from implicit feedback*; Semi-supervised learning settings.

KEYWORDS

Cross-Domain Recommendation; Semi-Supervised Learning; Collaborative Filtering; Metric Learning; Neighborhood Inference

ACM Reference Format:
SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357914

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357914

1 INTRODUCTION

As an increasing number of users interact with one or more their interest domains, recent recommender systems (RS) started to leverage user-item interactions (e.g., ratings or feedbacks) collected from multiple domains to improve the quality of recommendations, so-called cross-domain recommendation (CDR) [5–7, 11, 12, 18, 19]. CDR aims to transfer the knowledge acquired from a domain to another domain, and this inter-domain information exchange is quite helpful to combat the data sparsity problem, which is one of the long-standing problems in conventional RS.

Most CDR methods [7, 11, 16, 27] are mainly interested in improving the overall recommendation accuracy for a target domain by the help of the users who also exist in other domains (i.e., *overlapping users*). Thus, they cannot make recommendations to the users who are newly joined or have not interacted with any items yet (i.e., *cold-start users*). This is another important task in CDR, and it is more technically challenging as well as has great values from a practical perspective [19]. To this end, RS should be able to infer the preferences of such cold-start users based on their preferences observed in other domains. For example, providers can recommend video-domain items to the users who want to try videos for the first time by analyzing their preferences in a book-domain. In this paper, we use the term "target-domain" for the domain that we try to recommend, and "source-domain" for the domain that we transfer the knowledge of cold-start users' preferences from.

To tackle this challenge, several recent work [5, 18] focus on the Embedding and Mapping approach (EMCDR) which matches two latent spaces, where users' preferences on items are encoded respectively for a source-domain and a target-domain, by using the overlapping users of the two domains as anchor points. It trains a mapping function from the source-domain into the target-domain in a supervised manner so that it minimizes the distance between the actual latent vector in the target-domain and the approximated latent vector transferred from the source-domain for each overlapping user. After inferring the latent vectors of cold-start users by the trained mapping function, EMCDR is able to provide recommendations to the cold-start users.

However, existing EMCDR-based methods [5, 18, 27] have a limited performance in real-world CDR scenarios for the following reasons. 1) The EMCDR approach is only based on a supervised loss, which makes the mapping function biased to a small number of the overlapping users who are used as the anchors for training. From our in-depth analysis on the Amazon dataset, we found that the average ratio of the overlapping users to total users of any two domains is less than 5%. However, all the methods were evaluated in the experimental settings where a lot of users (up to 90%) are overlapped [5, 6, 12, 18], which distorts the real-world distribution.

2) They basically model the user-item interaction as the *inner-product* of their latent vectors, which makes a latent space violate triangle inequality [23]; in such spaces, the similarities among users and among items are not correctly captured [2, 10]. It eventually degrades the performance of collaborative filtering for each domain and also that of the cross-domain mapping because their mapping function is optimized by the *distance*-based supervised loss.

In this paper, we propose a Semi-Supervised framework for Cross-Domain Recommendation to cold-start users (SSCDR) that works well even in the case that there are only a few overlapping users in both the domains. To effectively capture the cross-domain relationship, SSCDR first learns the latent vectors of users and items in *metric spaces* to capture user-user similarities as well as user-item interactions in their distances. Then, it trains a cross-domain mapping function based on a *semi-supervised approach*, which optimizes the function to approximate the ground-truths of overlapping users (*labeled data*), and to encode their interactions with items simultaneously (*unlabeled data*). In addition, we develop a novel technique to accurately infer the latent vectors of the cold-start users considering the neighborhood of the users.

Our extensive experiments demonstrate that SSCDR outperforms all other baselines in terms of both the recommendation accuracy and the quality of cross-domain latent vector mapping. SSCDR shows the best CDR accuracy for cold-start users even in case that the number of overlapping users is very small, which follows a realistic data distribution, whereas the performances of the other methods drastically drop. Furthermore, we quantitatively validate that SSCDR is capable of inferring the latent vectors of cold-start users more accurately than the other methods, and qualitatively compare the target latent spaces induced by SSCDR and the others.

2 PRELIMINARIES

We first formally describe the notations and our problem (Section 2.1), then review an embedding and mapping approach which is a conventional strategy for cross-domain recommendation (CDR) (Section 2.2).

2.1 Notations and Problem Formulation

Suppose that we have two domains: a source-domain and a target-domain. In this paper, we use the term "target-domain" for the domain that we try to recommend, and "source-domain" for the domain that we transfer the knowledge of users' preferences from. We use the CDR scenario that "Book" is the source-domain and "Video" is the target-domain as an example case. Let $\mathcal{U}^s$, $\mathcal{U}^t$, $\mathcal{I}^s$, and $\mathcal{I}^t$ be the set of users and items in the source-domain and the target-domain, respectively. In most real-world cases, $\mathcal{U}^s$ and $\mathcal{U}^t$ are partially overlapped: the users who watch videos and also purchase books are a part of the total users. Thus, we define the set of overlapping users between the two domains by $\mathcal{OU} = \mathcal{U}^s \cap \mathcal{U}^t$. In contrast, $\mathcal{I}^s$ and $\mathcal{I}^t$ are disjoint sets, so there is no item that both the domains have in common. Given collaborative filtering (CF) information (i.e., implicit interactions between users and items) for each domain, we build a binary matrix that represents user-item interactions. The binary matrix $\boldsymbol{R}^s \in \{0,1\}^{|\mathcal{U}^s|\times|\mathcal{I}^s|}$ includes user-book purchase interactions, and similarly, $\boldsymbol{R}^t \in \{0,1\}^{|\mathcal{U}^t|\times|\mathcal{I}^t|}$ includes user-video watching interactions. We additionally use the notations $\mathcal{NI}_i^s$ for the set of the items that the user i has interacted with in the source-domain (i.e., the neighbor items of the user i), and $\mathcal{NU}_j^s$ for the set of the users that previously interacted with the item j in the source-domain (i.e., the neighbor users of the item j). Our goal is to recommend top-N items in $\mathcal{I}^t$ to *cold-start users* who have not interacted with $\mathcal{I}^t$ at all. In other words, we aim to recommend videos to users who have never watched videos but only purchased books.

2.2 Embedding and Mapping Approach

The Embedding and Mapping approach for Cross-Domain Recommendation (EMCDR), which is a general CDR strategy for cold-start users, is firstly proposed in [18], and several methods have been proposed based on this approach [5, 27]. EMCDR consists of 1) the embedding step for modeling users and items in latent spaces, and 2) the mapping step for learning the inter-domain relationship.

In the embedding step, EMCDR learns the user and item latent vectors for each domain by using Matrix Factorization (MF). MF models the user-item interaction as the inner product of their latent vectors [15]. Given a dataset that contains implicit interactions between users and items, the latent vectors are optimized based on the assumption that a user prefers interacted items to non-interacted items [24]. This can be formalized as the following pairwise loss:

$$\min_{\mathbf{u}*,\mathbf{v}*} \sum_{i\in\mathcal{U}} \sum_{j\in\mathcal{NI}_i} \sum_{k\notin\mathcal{NI}_i} -\log\sigma(\mathbf{u}_i^\top\mathbf{v}_j - \mathbf{u}_i^\top\mathbf{v}_k),$$

where σ is the sigmoid function, $\mathbf{u}_* \in \mathbb{R}^d$, $\mathbf{v}_* \in \mathbb{R}^d$ are user and item latent vectors, and d is the dimension size of the latent vectors. In this way, EMCDR learns the latent matrices $\boldsymbol{U}^s \in \mathbb{R}^{|\mathcal{U}^s|\times d}$, $\boldsymbol{V}^s \in \mathbb{R}^{|\mathcal{I}^s|\times d}$ from $\boldsymbol{R}^s$, and $\boldsymbol{U}^t \in \mathbb{R}^{|\mathcal{U}^t|\times d}$, $\boldsymbol{V}^t \in \mathbb{R}^{|\mathcal{I}^t|\times d}$ from $\boldsymbol{R}^t$.

In the mapping step, EMCDR learns a mapping function f_θ to capture the inter-domain relationship by utilizing the overlapping users as a bridge. The learning procedure is formalized as a supervised regression problem,

$$\min_\theta \sum_{i\in\mathcal{OU}} ||f_\theta\left(\mathbf{u}_i^s\right) - \mathbf{u}_i^t||^2.$$

The mapping function is optimized to minimize the distance between the user latent vectors transferred from the source-domain by cross-domain mapping and the corresponding user latent vectors in the target-domain. After training the mapping function, EMCDR can infer the target-domain latent vector of a cold-start user i by $\hat{\mathbf{u}}_i^t = f_\theta(\mathbf{u}_i^s)$. Using the inferred latent vector and the target-domain item vectors $\boldsymbol{V}^t$, EMCDR provides recommendations to the user.

The existing EMCDR-based methods have a limited performance for the following reasons. First, the training of the mapping function only uses the supervised loss, whose performance is sensitive to the number of overlapping users. In particular, in case that there are a small number of the overlapping users, the mapping function is severely biased to them and fails to be generalized to the cold-start users. Second, they learn the user-item interaction as the inner product of their latent vectors, which makes the latent space violate *triangle inequality* [23]. As pointed out in [2, 10], the user-user and item-item similarities are not accurately modeled in such space. In addition, the supervised loss for the mapping function defined by the Euclidean distance is not fully compatible with their latent space, because the user-item interactions are captured in the space as the inner product, not the distance.

3 REAL-WORLD CDR SETTING ANALYSIS

In this section, based on our detailed analyses on the Amazon dataset[1], we clarify the challenge of the CDR task and the limitation of existing CDR methods. The Amazon dataset is the largest public CDR dataset, which contains more than 1.3 million users and 42 item domains.

First, we analyze two factors that have a significant impact on the CDR performance for all possible cross-domain scenarios, determined by any two domains.

$$R_{\text{oou}} = \frac{|\mathcal{OU}|}{|\mathcal{U}^s \cup \mathcal{U}^t|}, \quad R_{\text{oos}} = \frac{|\mathcal{OU}|}{|\mathcal{U}^s|}$$

The first factor is the ratio of the overlapping users over the union of the source-domain and the target-domain users, denoted by R_{oou}. As CDR methods usually use the overlapping users as a bridge to transfer knowledge between two domains, the overall performances could be improved as R_{oou} increases. The second factor is the ratio of the overlapping users over the source-domain users, denoted by R_{oos}. The higher R_{oos}, the more information of the source-domain can be used to learn the cross-domain relationship, which results in more accurate transfer of users in the source-domain. Figure 1 shows the histogram of R_{oou} and R_{oos}. We observe that cross-domain scenarios with R_{oou} less than 10% account for 92% of the total scenarios. Similarly, the scenarios with R_{oos} less than 20% account for 83% of the total scenarios.

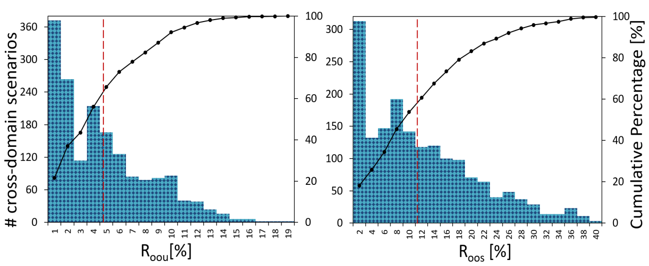


Figure 1: Histograms of R_{oou} and R_{oos} on cross-domain scenarios (Red dotted lines: the averages)

We also compare the experimental settings used in the recent CDR methods for cold-start users, including our proposed method. In Table 1, all the previous work have evaluated their methods in the experimental settings where both R_{oou} and R_{oos} range from 20% to 90%. Most of them first start from the case that all users are *completely overlapped* between the two domains and gradually remove the users in the target-domain. However, as shown in Figure 1, scenarios with R_{oou} larger than 20% rarely exist, and scenarios with R_{oos} larger than 20% only account for 19% of the total scenarios. Although our data analyses do not cover all real-world CDR scenarios, at least we confirm that the existing methods have proved their effectiveness in the experimental environments that are distant from reality.

[1] https://registry.opendata.aws/amazon-reviews/

Table 1: Experimental settings (training set) of recent CDR methods for cold-start users and SSCDR

Methods	R_{oou}	R_{oos}	Side Information
GCBAN [6]	20 ∼ 80%	20 ∼ 80%	Yes
RC-DFM [5]	20 ∼ 80%	20 ∼ 80%	Yes
EMCDR [18]	50 ∼ 90%	50 ∼ 90%	No
SSCDR	0.2 ∼ 5.6%	0.3 ∼ 21.2%	No

4 THE SSCDR FRAMEWORK

In this section, we provide the details of SSCDR framework which bridges semi-supervised learning to CDR. Our proposed framework consists of the following three major steps. In the first step, SSCDR learns user and item latent vectors in metric spaces for the source-domain and target-domain, respectively (Section 4.1). In the second step, SSCDR trains a mapping function that captures the cross-domain relationships between the two metric spaces (Section 4.2). In the third step, SSCDR infers the latent vectors of cold-start users based on their neighborhood in the source-domain, then recommends the target-domain items to cold-start users (Section 4.3). Algorithm 1 describes the overall process of our framework.

4.1 Collaborative Filtering in Metric Space

The first step of the SSCDR framework is to learn a metric space for each domain. To be specific, we embed users and items into the low-dimensional metric space where a user's preference on an item is inversely proportional to their Euclidean distance. The distance between a user i and an item j is defined as

$$d(\mathbf{u}_i, \mathbf{v}_j) = \left\|\mathbf{u}_i - \mathbf{v}_j\right\|^2. \tag{1}$$

We optimize the space so that the distance between a user and its interacted items becomes smaller than the distance between the user and its non-interacted items. This collaborative filtering in metric space [10] is formulated by the following loss function:

$$\begin{aligned} \mathcal{L} = \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{NI}_i} \sum_{k \notin \mathcal{NI}_i} \left[m + d\left(\mathbf{u}_i, \mathbf{v}_j\right) - d\left(\mathbf{u}_i, \mathbf{v}_k\right)\right]_+, \\ s.t. \quad ||\mathbf{u}_*||^2 \leq 1 \text{ and } ||\mathbf{v}_*||^2 \leq 1, \end{aligned} \tag{2}$$

where m is margin size and $[x]_+ = \max(x, 0)$ is the standard hinge loss. We also add the unit sphere constraint to prevent user and item vectors from spreading too widely. This constraint is enforced by normalizing latent vectors (i.e., $\mathbf{u}_* \leftarrow \mathbf{u}_*/\max(1, \|\mathbf{u}_*\|_2)$, $\mathbf{v}_* \leftarrow \mathbf{v}_*/\max(1, \|\mathbf{v}_*\|_2)$, as done in [10].

Discussion: the necessity of metric space for CDR. The fundamental motivation of CDR is that users with similar tastes are located close to each other in the source-domain latent space, and they are likely to be located close in the target-domain latent space as well. For this reason, the latent space for each domain should be able to capture user-user similarities as well as user-item interactions. In the metric space that satisfies the triangle inequality, the user-user and item-item similarities are also captured with user-item interactions, because the users that share many interacted items (or the items that share many interacted users) get closer in the metric space [10]. To this end, we model users and items in the metric space, which is also well compatible with the cross-domain mapping in the following subsections.

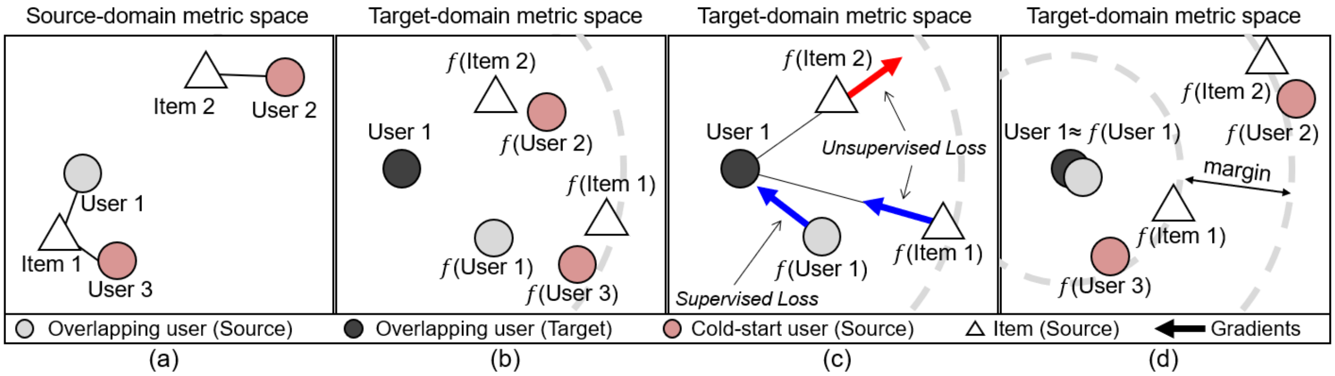


Figure 2: Illustration of the proposed semi-supervised mapping approach. (a) Learned metric space for the source-domain (solid lines: observed user-item interactions). (b) Transferred locations are confined to the overlapping user 1. (c) The supervised loss in Eq. (3) creates a gradient that pulls the transferred user 1 to the user 1. The unsupervised loss in Eq. (4) creates a gradient that pulls positive item 1 closer to the user 1 and pushes negative item 2. (d) Transferred locations of user 1, item 1 and item 2 are adjusted. As a result, transferred locations of user 2 and user 3 are naturally adjusted.

Algorithm 1: The SSCDR framework

Require: The interaction matrices, user and item sets of each domain $\boldsymbol{R}^s$, $\boldsymbol{R}^t$, $\mathcal{U}^s$, $\mathcal{U}^t$, $\mathcal{V}^s$, and $\mathcal{V}^t$

Ensure: Recommend top-N target-domain items to cold-start users

(1) Collaborative Filtering in Metric Space

1 Learn $\{\boldsymbol{U}^s, \boldsymbol{V}^s\}$ from $\boldsymbol{R}^s$ in a metric space

2 Learn $\{\boldsymbol{U}^t, \boldsymbol{V}^t\}$ from $\boldsymbol{R}^t$ in a metric space

(2) Semi-Supervised Metric Space Mapping

3 Learn the mapping function $f_\theta(\cdot)$ between two metric spaces

(3) Multi-hop Neighborhood Inference

4 Compute neighborhood-aggregated user latent vectors $\boldsymbol{U}^{s,H}$

5 Infer the latent vectors of cold-start users $\hat{\boldsymbol{U}}^t$

6 Make recommendations for the cold-start users

4.2 Semi-Supervised Metric Space Mapping

The second step of SSCDR is to learn a mapping function that captures the relationship between the two metric spaces (i.e., target-domain metric space and source-domain metric space). To effectively learn the relationship even with a small number of overlapping users, we employ the idea of semi-supervised learning. In general, semi-supervised learning [25, 28] aims at improving the overall performance of a supervised task when there are a small amount of labeled data and a large amount of unlabeled data. The underlying concept of the semi-supervised learning is based on the local-consistency assumption, which means data samples with high similarity are likely to have the same label. This assumption aligns with the fundamental motivation of CDR.

We formulate the training of cross-domain mapping function as a semi-supervised task. We regard the overlapping users as the labeled data, because we have the ground-truths about the target-domain locations where their source-domain latent vectors should be transferred by the mapping function. Without loss of generality, we can regard source-domain items that do not have ground-truths in the target-domain metric space as unlabeled data.

Supervised Loss. The supervised loss uses the overlapping users as the labeled data. The mapping function takes the user latent vectors in the source-domain as its inputs and the user latent vectors in the target-domain as the desired outputs. The loss function can be described by

$$\mathcal{L}_S = \sum_{i \in \mathcal{OU}} d\left(f_\theta\left(\mathbf{u}_i^s\right), \mathbf{u}_i^t\right), \tag{3}$$

where $f_\theta(\cdot)$ is the cross-domain mapping function, parameterized by θ. Note that the supervised loss tunes the location of each source-domain user vector transferred by the mapping function close to its ground-truth location in the target-domain metric space.

Unsupervised Loss. We design an effective unsupervised loss to involve the source-domain items into the training of the mapping function. Although the ground-truth locations of the source-domain items do not exist in the target-domain metric space, the mapping function can further learn the inter-domain relationship so that it transfers CF information observed in the source-domain (i.e., user-item interactions) to the target-domain metric space. Concretely, we transfer the latent vectors of the source-domain items to the target-domain metric space through the mapping function, then adjust the transferred locations according to whether it has interacted with the overlapping users or not. Based on this idea, we define the following unsupervised loss.

$$\mathcal{L}_U = \sum_{i \in \mathcal{OU}} \sum_{j \in \mathcal{NI}_i^s} \sum_{k \notin \mathcal{NI}_i^s} \left[m + d\left(f_\theta(\mathbf{v}_j^s), \mathbf{u}_i^t\right) - d\left(f_\theta(\mathbf{v}_k^s), \mathbf{u}_i^t\right)\right]_+, \tag{4}$$

where $f_\theta(\mathbf{v}_j^s)$ is the transferred location of source-domain item j, $[x]_+ = \max(x, 0)$ is the standard hinge loss, and m is the margin size. This loss ensures that the distance of an observed user-item pair is smaller than that of an unobserved pair by a margin of m in the target-domain metric space. In other words, each overlapping user pulls (or pushes) its interacted (or non-interacted) source-domain items in the target-domain metric space. The locations of the transferred item vectors are adjusted by the unsupervised loss, and the changes are propagated to their nearby users. That is, the transferred locations of cold-start users who have interacted with the items in the source-domain are also adjusted in the target-domain metric space.

The complete Loss function. The complete loss function for metric space mapping can be described as

$$\min_{\theta} \mathcal{L}_S + \lambda \cdot \mathcal{L}_U$$
$$s.t. \quad ||f_\theta(\mathbf{u}_*)||^2 \leq 1 \text{ and } ||f_\theta(\mathbf{v}_*)||^2 \leq 1. \quad (5)$$

λ is the weight hyperparameter that controls the importance of the unsupervised loss, and θ denotes the parameters of the mapping function. The unit-sphere constraint is also enforced in the same way with Section 4.1.

Discussion: the effect of the semi-supervised loss. In case that the mapping function is trained by the supervised loss, the locations of users and items transferred by the mapping function are easily confined near to the overlapping users. Figure 2 illustrates how the semi-supervised mapping approach alleviates this problem. In our complete loss function, the supervised loss (Eq. (3)) learns the labeled data as anchor points, and the unsupervised loss (Eq. (4)) makes the unlabeled data spread over the target-domain metric space according to their relative similarities to the anchors. Our metric spaces correctly encode the similarity among entities (i.e., user and item), so the unlabeled source-domain items also can be involved in the training of the mapping function based on their similarity to the overlapping users. As a result, the transferred locations of cold-start users are no longer biased to the overlapping users and get closer to the points that represent their preferences accurately.

4.3 Multi-hop Neighborhood Inference

The final step of SSCDR is to recommend target-domain items to cold-start users. We propose an inference technique that accurately infers the latent vectors of the cold-start users by fully utilizing their multi-hop neighborhood[2] information (i.e., a set of items interacted with a user, and a set of users interacted with an item) [20, 26]. We first compute new vectors of the cold-start users in the source-domain metric space by repeatedly aggregating their neighbors, then transfer the vectors to the target-domain metric space by using the mapping function.

The h-th aggregated latent vectors of a user i and an item j in the source-domain metric space can be represented as

$$\mathbf{v}_j^{s,h} = \frac{1}{|\mathcal{NU}_j^s| + 1}\left(\mathbf{v}_j^{s,h-1} + \sum_{i \in \mathcal{NU}_j^s} \mathbf{u}_i^{s,h-1}\right), \quad (6)$$

$$\mathbf{u}_i^{s,h} = \frac{1}{|\mathcal{NI}_i^s| + 1}\left(\mathbf{u}_i^{s,h-1} + \sum_{j \in \mathcal{NI}_i^s} \mathbf{v}_j^{s,h-1}\right). \quad (7)$$

We set the initial vectors (in case of $h = 0$) to the learned user and item vectors; $\mathbf{v}_j^{s,0} = \mathbf{v}_j^s, \mathbf{u}_i^{s,0} = \mathbf{u}_i^s$. In this way, each final H-th aggregated user vector encompasses the information of both the user and its H-hop neighbors. Finally, SSCDR infers the latent vector of a cold-start user i by

$$\hat{\mathbf{u}}_i^t = f_\theta(\mathbf{u}_i^{s,H}), \quad (8)$$

and recommend the top-N items, which are located nearest to the latent vector in the target-domain metric space, to the user.

[2]We use the term "neighbors" to represent the set of each user's interacted items and the set of each item's interacted users, rather than the nearest entities in a space.

Discussion: the effect of the multi-hop neighborhood inference. The naive inference of the cold-start users' preferences, which transfers their source-domain latent vectors to the target-domain by the trained mapping function, has a limited performance. This is because their interactions in the source-domain are only considered implicitly (i.e., captured in their source-domain latent vectors). To improve the accuracy of the inference, our proposed inference technique predicts the locations of the cold-start users by explicitly making use of their interactions (i.e., neighbors). Their neighborhood information in the source-domain is helpful for the accurate inference, because our mapping function is trained to capture user-item interactions observed in the source-domain by the semi-supervised mapping approach.

5 EXPERIMENTS

We conduct extensive experiments to evaluate our SSCDR framework. We first show the superior performance of SSCDR over the state-of-the-art methods (Section 5.2), and analyze the effectiveness of each component in SSCDR framework in the target-domain metric space (Section 5.3). We also investigate the performance changes of SSCDR with respect to its hyperparameters (Section 5.4).

5.1 Experimental Settings

Table 2: Statistics of the cross-domain scenarios (Inter. denotes interactions and Overlap. denotes overlapping users)

		#Users	#Items	#Inter.	Density	#Overlap.
Scenario 1	Book	4,291	18,984	171,206	0.21%	1,823
	Video	17,690	48,376	883,226	0.10%	
Scenario 2	Music	8,783	35,672	427,112	0.14%	1,306
	Book	4,090	18,920	168,317	0.22%	
Scenario 3	Video	17,902	48,402	886,247	0.10%	2,029
	Game	3,680	12,416	104,465	0.23%	

Dataset. We use the Amazon dataset[3] to evaluate our model and baselines. The dataset contains 42 different item domains, and among them, we choose the four popular categories: Music, Book, Video, and Game. We regard each observed rating as an implicit feedback record (i.e., user-item interaction) as done in [8, 9, 21]. Then, we define three CDR scenarios as *Scenario 1*: Book → Video, *Scenario 2*: Music → Book, and *Scenario 3*: Video → Game. For each scenario, we filter out the overlapping users whose number of interacted items is fewer than 10, and also filter out non-overlapping users and items with fewer than 20 interactions [5, 8, 10, 13, 24]. The details of our CDR scenarios are summarized in Table 2.

Experimental Setup. For each scenario, we randomly select 50% of the total overlapping users and remove their information in the target-domain in order to utilize them as cold-start users for calculating the recommendation accuracy (i.e., test users). To investigate the performance changes of CDR methods with respect to the number of overlapping users, we restrict the number of the overlapping users similarly to the real-world distribution as presented in Section 3. For each scenario, we build four training sets by randomly including only a certain fraction $\phi \in \{5\%, 10\%, 50\%, 100\%\}$ of the overlapping users who do not belong to the test users.

[3]This dataset is the same with the one analyzed in Section 3.

Evaluation Protocols. We follow the *leave-one-out* evaluation protocol widely used in the Top-N recommendation task [8, 9, 11, 21]. For each test user, we leave out a single interacted items for testing, and use the rest for training. In our experiments, we leave out an additional interacted item for the validation. To address the time-consuming issue of ranking all the items, we randomly sample 999 items among the items that have not interacted with the user (i.e., negative items), then evaluate how well each method can rank the test item higher than these negative items. We repeat this process of sampling a test/validation item and negative items five times and report the average results. As we focus on the top-N recommendation task based on implicit feedback, we evaluate the performance of each method by using the three ranking metrics: hit ratio (HR), normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR). The HR H@N simply measures whether the test item is present in the top-N list:

$$\mathrm{H@}N = \frac{1}{|\mathcal{U}_{test}|} \sum_{i \in \mathcal{U}_{test}} \delta\,(p_i \leq \text{ top } N)\,,$$

where $\mathcal{U}_{test}$ is the set of the test users, p_i is the hit position of the test item for the user i, and $\delta(\cdot)$ is the indicator function. The NDCG N@N and the MRR M@N are position-aware ranking metrics that assign higher scores to the hits at upper ranks:

$$\mathrm{N@}N = \frac{1}{|\mathcal{U}_{test}|} \sum_{i \in \mathcal{U}_{test}} \frac{\log 2}{\log\,(p_i + 1)},\ \mathrm{M@}N = \frac{1}{|\mathcal{U}_{test}|} \sum_{i \in \mathcal{U}_{test}} \frac{1}{p_i}$$

Baselines. We categorize our baseline methods into three groups according to their approaches. A method in the first group performs *non-personalized* recommendations (i.e., a non-CF method).

- **ITEMPOP**: A simple method which makes a ranked list of all items in the target-domain by the number of their interactions (i.e., popularity). This method provides benchmark performances of the non-personalized recommendation.

The methods in the second group consider all domains as a single domain for CDR [13, 19]. We construct a unified matrix in a collective manner so that it takes all users and items as its rows and columns, respectively. Then, we apply the state-of-the-art CF methods designed for a single domain.

- **BPR [24]:** Bayesian personalized ranking method for implicit feedback datasets. It assumes that the items interacted with a user are more preferred than the items not interacted yet, and learns the pairwise ranking loss for latent vector modeling. The loss optimizes the order of scores defined by the inner product of user and item latent vectors.
- **CML [10]:** The state-of-the-art CF method for implicit feedback datasets based on a metric learning approach. It learns a joint metric space where users and items are embedded, and their distance is considered inversely proportional to the strength of the user's preference on the item.

The third group includes the state-of-the-art CDR methods for cold-start users.

- **CBMF [19]:** Clustering-based matrix factorization for top-N recommendation to cold-start users. Based on matrix factorization (MF), it learns and utilizes latent vectors at the two-levels: 1) latent vectors of each user and item, 2) latent vectors of each user-cluster and item-cluster.
- **EMCDR-BPR [18]:** Our main competitor which is the state-of-the-art CDR framework for cold-start users. It learns a cross-domain mapping function after latent vector modeling by BPR.[4]

Finally, we build three variants of SSCDR by excluding 1) the semi-supervised loss and 2) the inference technique to validate the effectiveness of each proposed component.

- **EMCDR-CML:** An ablation of SSCDR. It neither uses the semi-supervised approach nor the inference technique. Unlike the previous EMCDR that uses BPR for latent vector modeling, it applies CML to take advantages of the metric space.
- **$\mathbf{SSCDR_{naive}}$** and **SSCDR:** The proposed framework which trains a cross-domain mapping function based on the semi-supervised loss. $\mathrm{SSCDR_{naive}}$ straightforwardly infers the cold-start users by the mapping function (i.e., uses the naive inference), whereas SSCDR adopts the multi-hop neighborhood inference technique.

Note that there are several CDR methods designed to recommend items only to the users who already have interacted items in the target-domain [7, 11, 16, 27]; they are not included in our baselines because they cannot provide recommendations to cold-start users. In addition, there exist the CDR methods that make use of side information such as demographic profiles of users (e.g. age, gender, and profession) [6], review texts [5, 12], and item contents (e.g. plot and title of movie) [5, 6]; they are also not included in our baselines as they require such side information for CDR.

Implementation Details. We implement our framework and all the baselines using Tensorflow [1] for efficient computations on GPU, and use the Adam optimizer [14] to train all the methods. For the cross-domain mapping functions of EMCDR-BPR, EMCDR-CML, $\mathrm{SSCDR_{naive}}$ and SSCDR, we employ Multi-layer perceptron (MLP) which is mainly used in CDR methods [5, 18, 27], and apply the same network configuration as suggested in [18]. Their hidden layers have the shape of $[K \rightarrow 2 \times K \rightarrow K]$, and the *tanh* is used as the activation function.

For each scenario and method, the hyperparameters are tuned by grid searches using the validation set: the initial learning rate for the Adam optimizer in {0.1, 0.03 ,0.01, 0.003, 0.001, 0.0003, 0.0001}, the dimension size of the latent vector in {10, 30, 50, 100}, the model regularizer (for BPR and CBMF) in {0.01, 0.005, 0.001, 0.0005, 0.0001}, the margin size (for CML-based methods) in {0.5, 1.0, 1.5}, the weight parameter (for CBMF) in {0.2, 0.4, 0.6, 0.8, 1.0} and the number of clusters for users and items (for CBMF) is set to 100 which is provided by [19]. The hyperparameters for $\mathrm{SSCDR_{naive}}$ and SSCDR are tuned in the range of $\lambda \in \{0.25, 0.5, 0.75, 1.0\}$ and $H \in \{1,2,3,4\}$.

5.2 Comparing Different Approaches

We quantitatively evaluate the recommendation performance of SSCDR and other methods in terms of three ranking metrics. Table 3 shows the results on three scenarios with four different ϕ values, which are the fractions of the overlapping users. In summary, for all scenarios and ϕ, SSCDR achieves the best performance among all the baselines, and specifically, it shows the significant improvement (min: 6%, max: 41%, avg: 17%) in terms of H@10 compared to the

[4] In case of a rating prediction task, probabilistic matrix factorization (PMF) is used instead of BPR in [18], but we focus on top-N ranking task in this paper.

Table 3: Recommendation performances. Best results are in bold face. $*$, $**$, and $***$ indicate $p \leq 0.05$, $p \leq 0.01$, and $p \leq 0.005$, paired t-test of SSCDR vs. the best baselines on H@10. *Improv. over sup.* denotes the improvement of SSCDR over EMCDR-CML.

	ϕ	Metrics	ITEMPOP	BPR	CML	CBMF	EMCDR-BPR	EMCDR-CML	SSCDR$_{\text{naive}}$	SSCDR	*Improv. over best*	*Improv. over sup.*
Scenario 1	5%**	H@10	0.1267	0.1412	0.1434	0.1425	0.1994	0.2068	0.2285	**0.2354**	18.05%	13.83%
		H@20	0.2029	0.2220	0.2226	0.2141	0.2959	0.2979	0.3238	**0.3313**	11.96%	11.21%
		N@10	0.0646	0.0710	0.0737	0.0768	0.1079	0.1126	0.1247	**0.1273**	17.98%	13.06%
		N@20	0.0836	0.0930	0.0949	0.0946	0.1304	0.1343	0.1435	**0.1531**	17.41%	14.00%
		M@10	0.0459	0.0501	0.0527	0.0571	0.0801	0.0838	0.0932	**0.0949**	18.48%	13.25%
		M@20	0.0510	0.0576	0.0599	0.0617	0.0843	0.0888	0.0931	**0.1033**	22.54%	16.33%
	10%*	H@10	0.1267	0.1614	0.1636	0.1679	0.2178	0.2184	0.2329	**0.2362**	8.45%	8.15%
		H@20	0.2029	0.2481	0.2562	0.2505	0.3181	0.3144	0.3311	**0.3341**	5.03%	6.27%
		N@10	0.0646	0.0821	0.0822	0.0882	0.1169	0.1199	0.1265	**0.1286**	10.01%	7.26%
		N@20	0.0836	0.1012	0.1022	0.1071	0.1415	0.1435	0.1515	**0.1529**	8.06%	6.55%
		M@10	0.0459	0.0578	0.0593	0.0640	0.0862	0.0897	0.0943	**0.0968**	12.30%	7.92%
		M@20	0.0510	0.0610	0.0624	0.0673	0.0921	0.0953	0.1023	**0.1027**	11.51%	7.76%
	50%*	H@10	0.1267	0.2015	0.2053	0.2083	0.2351	0.2345	0.2498	**0.2536**	7.87%	8.14%
		H@20	0.2029	0.2959	0.3032	0.2913	0.3390	0.3313	0.3526	**0.3550**	4.72%	7.15%
		N@10	0.0646	0.1104	0.1112	0.1110	0.1285	0.1315	0.1393	**0.1404**	9.26%	6.77%
		N@20	0.0836	0.1298	0.1308	0.1322	0.1570	0.1548	0.1626	**0.1655**	5.41%	6.91%
		M@10	0.0459	0.0828	0.0827	0.0815	0.0960	0.1003	0.1057	**0.1064**	10.83%	6.08%
		M@20	0.0510	0.0841	0.0832	0.0879	0.1059	0.1048	0.1096	**0.1130**	6.70%	7.82%
	100%*	H@10	0.1267	0.2158	0.2228	0.2239	0.2413	0.2417	0.2549	**0.2562**	6.17%	6.00%
		H@20	0.2029	0.3102	0.3170	0.3181	0.3445	0.3410	0.3543	**0.3559**	3.31%	4.37%
		N@10	0.0646	0.1171	0.1198	0.1203	0.1341	0.1373	0.1410	**0.1451**	8.20%	5.68%
		N@20	0.0836	0.1403	0.1419	0.1428	0.1575	0.1602	0.1664	**0.1668**	5.90%	4.12%
		M@10	0.0459	0.0872	0.0887	0.0890	0.1015	0.1049	0.1064	**0.1115**	9.85%	6.29%
		M@20	0.0510	0.0932	0.0929	0.0942	0.1052	0.1089	0.1135	**0.1143**	8.65%	4.96%
Scenario 2	5%**	H@10	0.0988	0.0830	0.0918	0.0881	0.0834	0.0901	0.0976	**0.1143**	15.69%	26.86%
		H@20	0.1434	0.1295	0.1389	0.1246	0.1359	0.1409	0.1492	**0.1656**	15.48%	17.53%
		N@10	0.0557	0.0432	0.0451	0.0463	0.0434	0.0466	0.0482	**0.0580**	4.13%	24.46%
		N@20	0.0669	0.0546	0.0560	0.0557	0.0544	0.0572	0.0634	**0.0697**	4.19%	21.85%
		M@10	0.0409	0.0311	0.0310	0.0337	0.0314	0.0336	0.0334	**0.0410**	0.24%	22.02%
		M@20	**0.0439**	0.0331	0.0333	0.0366	0.0339	0.0345	0.0400	0.0434	-1.14%	25.80%
	10%*	H@10	0.0988	0.0914	0.0947	0.0934	0.0916	0.0967	0.1066	**0.1152**	16.60%	19.13%
		H@20	0.1434	0.1381	0.1422	0.1467	0.1418	0.1562	0.1651	**0.1770**	20.65%	13.32%
		N@10	0.0557	0.0456	0.0449	0.0475	0.0466	0.0505	0.0554	**0.0596**	7.00%	18.02%
		N@20	0.0669	0.0579	0.0584	0.0584	0.0596	0.0650	0.0681	**0.0728**	8.82%	12.00%
		M@10	0.0409	0.0323	0.0302	0.0336	0.0330	0.0366	0.0389	**0.0428**	4.65%	16.94%
		M@20	0.0439	0.0347	0.0356	0.0345	0.0362	0.0401	0.0417	**0.0449**	2.28%	11.97%
	50%*	H@10	0.0988	0.1230	0.1259	0.1238	0.1236	0.1262	0.1365	**0.1389**	10.33%	10.06%
		H@20	0.1434	0.1918	0.1951	0.1906	0.1911	0.1938	0.2041	**0.2070**	6.10%	6.81%
		N@10	0.0557	0.0603	0.0613	0.0619	0.0642	0.0646	0.0682	**0.0699**	8.88%	8.14%
		N@20	0.0669	0.0783	0.0798	0.0763	0.0790	0.0804	0.0859	**0.0862**	8.02%	7.19%
		M@10	0.0409	0.0416	0.0418	0.0433	0.0463	0.0467	0.0478	**0.0492**	6.26%	5.26%
		M@20	0.0439	0.0474	0.0481	0.0466	0.0477	0.0496	0.0526	**0.0531**	10.40%	7.14%
	100%*	H@10	0.0988	0.1283	0.1320	0.1291	0.1320	0.1336	0.1372	**0.1426**	8.03%	6.74%
		H@20	0.1434	0.1975	0.2061	0.1996	0.1967	0.2037	0.2135	**0.2184**	5.97%	7.22%
		N@10	0.0557	0.0631	0.0665	0.0652	0.0658	0.0680	0.0693	**0.0701**	5.41%	3.09%
		N@20	0.0669	0.0849	0.0872	0.0855	0.0851	0.0879	0.0905	**0.0909**	4.24%	3.41%
		M@10	0.0409	0.0436	0.0468	0.0465	0.0478	0.0488	0.0494	**0.0494**	3.35%	1.23%
		M@20	0.0439	0.0534	0.0545	0.0546	0.0542	0.0552	0.0559	**0.0565**	3.48%	2.36%
Scenario 3	5%***	H@10	0.0935	0.0830	0.0866	0.0876	0.0835	0.1024	0.1091	**0.1179**	26.10%	15.14%
		H@20	0.1540	0.1371	0.1391	0.1440	0.1367	0.1653	0.1712	**0.1817**	17.99%	9.92%
		N@10	0.0462	0.0398	0.0448	0.0418	0.0422	0.0532	0.0568	**0.0597**	29.22%	12.22%
		N@20	0.0613	0.0537	0.0572	0.0554	0.0576	0.0698	0.0723	**0.0759**	23.82%	8.74%
		M@10	0.0321	0.0270	0.0322	0.0354	0.0298	0.0385	0.0411	**0.0430**	21.47%	11.69%
		M@20	0.0362	0.0309	0.0347	0.0401	0.0350	0.0437	0.0452	**0.0473**	17.96%	8.24%
	10%***	H@10	0.0935	0.1065	0.1073	0.1105	0.1055	0.1177	0.1217	**0.1286**	16.38%	9.26%
		H@20	0.1540	0.1700	0.1744	0.1807	0.1639	0.1913	0.1963	**0.1996**	10.46%	4.34%
		N@10	0.0462	0.0509	0.0509	0.0527	0.0512	0.0611	0.0629	**0.0663**	25.81%	8.51%
		N@20	0.0613	0.0679	0.0720	0.0701	0.0686	0.0803	0.0817	**0.0817**	13.47%	1.74%
		M@10	0.0321	0.0343	0.0391	0.0354	0.0369	0.0440	0.0453	**0.0476**	21.74%	8.18%
		M@20	0.0362	0.0399	0.0439	0.0401	0.0424	0.0500	0.0504	**0.0526**	19.82%	5.20%
	50%*	H@10	0.0935	0.1272	0.1292	0.1302	0.1345	0.1358	0.1404	**0.1426**	6.02%	5.01%
		H@20	0.1540	0.2039	0.2022	0.2028	0.2065	0.2093	0.2162	**0.2199**	6.49%	5.06%
		N@10	0.0462	0.0648	0.0672	0.0660	0.0706	0.0715	0.0724	**0.0742**	5.10%	3.78%
		N@20	0.0613	0.0845	0.0846	0.0837	0.0906	0.0879	0.0912	**0.0923**	1.88%	5.01%
		M@10	0.0321	0.0461	0.0487	0.0467	0.0505	0.0513	0.0520	**0.0539**	6.73%	5.07%
		M@20	0.0362	0.0516	0.0525	0.0511	0.0551	0.0546	0.0569	**0.0578**	4.90%	5.86%
	100%*	H@10	0.0935	0.1353	0.1355	0.1385	0.1366	0.1406	0.1477	**0.1481**	6.93%	5.33%
		H@20	0.1540	0.2097	0.2172	0.2142	0.2136	0.2164	0.2239	**0.2252**	3.68%	4.07%
		N@10	0.0462	0.0683	0.0682	0.0694	0.0722	0.0740	0.0764	**0.0766**	6.09%	3.51%
		N@20	0.0613	0.0862	0.0907	0.0880	0.0901	0.0924	0.0951	**0.0965**	6.39%	4.44%
		M@10	0.0321	0.0482	0.0480	0.0488	0.0527	0.0538	0.0551	**0.0552**	4.74%	2.60%
		M@20	0.0362	0.0524	0.0562	0.0534	0.0566	0.0583	0.0598	**0.0615**	8.66%	5.49%

state-of-the-art method (i.e., EMCDR-BPR). We analyze the above results with various perspectives.

The impact of the number of overlapping users. We observe that the performances of the existing methods are severely limited by the number of overlapping users. Their performances drastically drop as the number of overlapping users decreases, and they even perform worse than the non-personalized recommendation method (i.e. ITEMPOP). Specifically, in case of Scenario 2 with $\phi = 5\%$ and 10%, every competitor shows worse performances than ITEMPOP. Likewise, in Scenario 3 with $\phi = 5\%$, all the competitors except EMCDR-CML are worse than ITEMPOP. In these settings, leveraging the knowledge obtained from another domain is no longer beneficial for the existing methods. However, in the same settings, SSCDR outperforms ITEMPOP by up to 26.10% (for H@10 in Scenario 3). These results show that the proposed framework is capable of effectively exploiting the cross-domain knowledge even in the case that there are only a few overlapping users, and successfully overcomes the limitation of the existing methods.

The effectiveness of the metric space. We observe that EMCDR-CML outperforms EMCDR-BPR, and CML outperforms BPR in general. This tendency is consistent with the previous work [10], which verified the benefits of learning a metric space where the triangle inequality is satisfied. This also emphasizes that the latent space should be able to correctly capture user-user similarity for cross-domain recommendation, as described in Section 4.1.

The effectiveness of the semi-supervised mapping. We observe that $\text{SSCDR}_{\text{naive}}$ significantly outperforms EMCDR-CML by up to 10.49% (for H@10 in Scenario 1 with $\phi = 5\%$). Notably, $\text{SSCDR}_{\text{naive}}$ recommends more accurately than EMCDR-CML that uses double the number of overlapping users (for H@10 in Scenario 1 and Scenario 2 with $\phi = 5\%$ (vs. 10%) and 50% (vs. 100%)). As $\text{SSCDR}_{\text{naive}}$ and EMCDR-CML have the same mapping network configuration, their performance gap results from the proposed semi-supervised metric space mapping. This results indicate that our semi-supervised approach achieves much better generalization compared to the supervised approach.

The effectiveness of the multi-hop neighborhood inference. The superior performance of SSCDR over $\text{SSCDR}_{\text{naive}}$ confirms that the proposed inference technique can infer more accurate latent vectors of the cold-start users than the naive inference, by effectively integrating their neighborhood information.

5.3 Target-Domain Metric Space Analysis

We conduct both qualitative and quantitative analyses on target-domain metric space to further investigate why SSCDR outperforms the state-of-the-art methods. For the qualitative analysis, we visualize the target-domain metric space by using t-distributed Stochastic Neighbor Embedding (t-SNE)[5] [17] to observe how the inferred locations of the cold-start users[6] are affected by the semi-supervised mapping and the multi-hop neighborhood inference. For the quantitative analysis, we measure the actual distances between the inferred latent vectors and their ground-truths.

[5]We employ the default setting of the t-SNE provided by Scikit-learn [22].

[6]The test users have no ground-truths in the target-domain metric space. Thus, in this experiment, we use the overlapping users of each scenario with $\phi = 100\%$, by splitting them into 5% for training the mapping function and the other 95% for testing with their ground-truths (i.e., cold-start users).

Figure 3 shows the target-domain metric space in case of Scenario 1. Red points represent the inferred latent vectors by EMCDR-CML, and green points represent the inferred latent vectors by SSCDR. Blue points represent ground-truth vectors of the cold-start users. Black points and translucent circles represent the overlapping users and their surrounding areas, respectively.

How does the semi-supervised loss affect the mapping function? (EMCDR-CML vs. $\textbf{SSCDR}_{\textbf{naive}}$) In Figure 3(a), the red points are mostly located within the translucent circles, showing that the mapping function is severely biased to the overlapping users. On the contrary, in Figure 3(b), the green points are not confined to the circles but scattered over the target-domain metric space. That is, the semi-supervised approach helps the mapping function not to be biased to the overlapping users and to achieve better generalization than the supervised approach.

How does the multi-hop neighborhood inference affect the transfer between two domains? ($\textbf{SSCDR}_{\textbf{naive}}$ vs. SSCDR) The green points in Figure 3(c) more overlap with the ground-truth blue points, compared to Figure 3(b). This implies that the inference technique effectively integrates neighborhood information to accurately predict the locations of the cold-start users. We can also observe the effect of different degrees of the aggregation (i.e., H) in Figure 3(c) and (d).

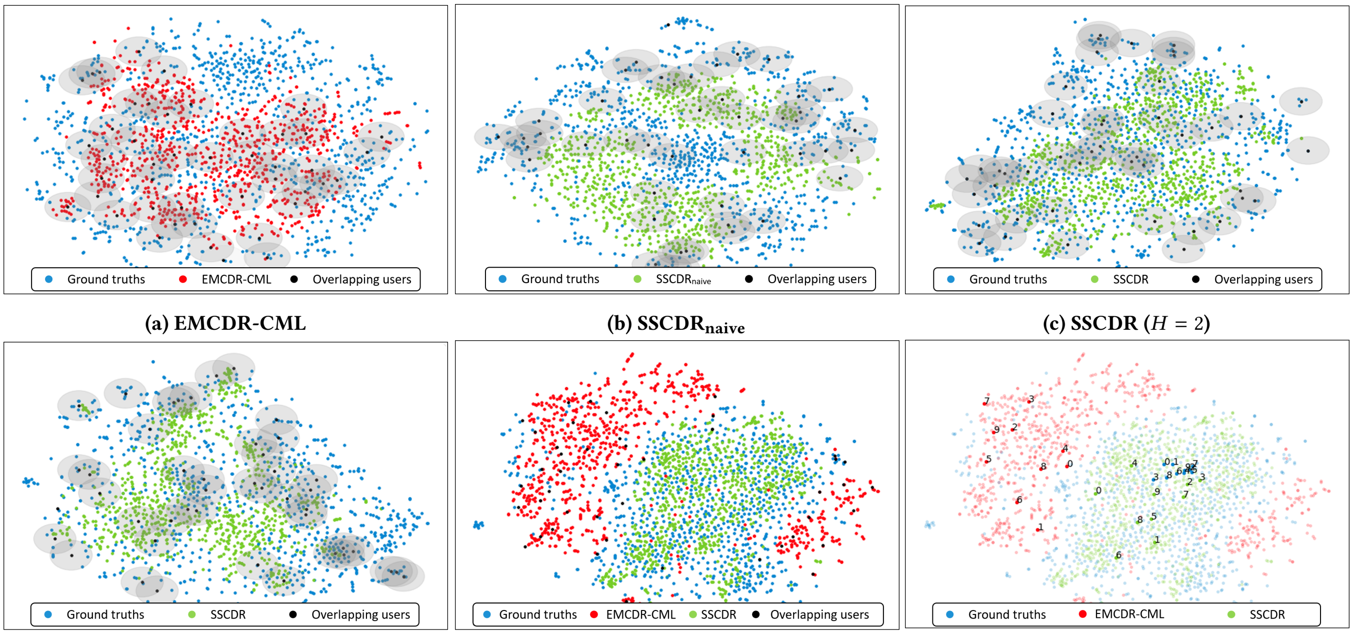
Does SSCDR indeed infer more accurate latent vectors of the cold-start users? (EMCDR-CML vs. SSCDR) In Figure 3(e), we visualize the latent vectors inferred by EMCDR-CML (red points) and SSCDR (green points) at once. As shown in Figure 3(a) and (c), the red points are clustered near the overlapping users (black points), whereas the green points are well mixed with the ground-truth points (blue points). Figure 3(f) provides the further analysis on ten users; the points with the same number represent the the same user. We confirm that SSCDR indeed transfers the cold-start users to more accurate locations than EMCDR-CML, and based on the inferred vectors, it can provide better recommendations.

Furthermore, we analyze the actual distances between the inferred latent vectors and their ground-truths for the cold-start users in all the scenarios in Figure 4. We observe that the latent vectors inferred by SSCDR are indeed located closer to their ground-truths than the latent vectors inferred by EMCDR-CML. Specifically, the average distances are reduced by up to 17.32% (for Scenario 3). This again verifies the effectiveness of the proposed framework.

5.4 Hyperparameter Analysis

We analyze the effect of two hyperparameters: λ and H. Table 4 shows the effect of λ on the recommendation performance, where λ denotes the importance weight of the unsupervised loss. Larger λ values imply a stronger contribution of the unsupervised loss to the mapping function, and $\lambda = 0$ indicates no contribution which makes our model identical to EMCDR-CML. To analyze only the effect of λ on the metric space mapping, we report the results without the inference technique (i.e., $\text{SSCDR}_{\text{naive}}$). We observe that the proposed unsupervised loss (in cases of $\lambda > 0$) is significantly beneficial to improve the model performance.

Similarly, Table 5 shows the effect of H on the performance of SSCDR, where H denotes the degree of neighborhood aggregation in our inference step. $H = 0$ is the case of SSCDR that adopts the



(a) EMCDR-CML (b) SSCDR$_{\text{naive}}$ (c) SSCDR ($H = 2$)

(d) SSCDR ($H = 4$) (e) EMCDR-CML and SSCDR ($H = 2$) (f) EMCDR-CML and SSCDR ($H = 2$)

Figure 3: t-SNE visualization of transferred user latent vectors in target-domain metric space

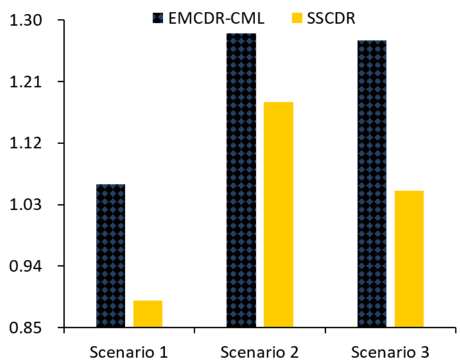


Figure 4: Average distance between the inferred user latent vectors and their ground-truths.

naive inference without the aggregation (i.e., SSCDR$_{\text{naive}}$). Larger H values mean the more neighborhood information is aggregated for the inference. We observe that aggregating the neighborhood information is helpful to accurately infer the latent vectors of the cold-start users, and the improvement becomes larger as fewer overlapping users are available.

6 RELATED WORK

We briefly survey the literature on CDR methods developed for two different target user groups: 1) target-domain users and 2) cold-start users.

CDR methods for target-domain users. Most CDR methods aim to improve the recommendation accuracy for the users in the target-domain [7, 11, 16, 27]. These methods integrate the knowledge acquired from the source-domain and the target-domain to improve the quality of recommendations in a collective manner.

Table 4: Effect of λ on H@10

Scenarios	ϕ	λ 0.0	0.25	0.50	0.75	1.0
Scenario 1	5%	0.2068	0.2221	0.2235	0.2272	**0.2285**
	10%	0.2184	0.2309	0.2322	**0.2329**	0.2318
	50%	0.2345	0.2493	0.2493	**0.2498**	0.2494
	100%	0.2417	**0.2549**	0.2529	0.2529	0.2525
Scenario 2	5%	0.0901	**0.0976**	0.0934	0.0905	0.0897
	10%	0.0967	**0.1066**	0.1028	0.1016	0.1004
	50%	0.1262	0.1266	0.1358	**0.1365**	0.1262
	100%	0.1336	0.1319	0.1364	0.1352	**0.1372**
Scenario 3	5%	0.1024	0.1082	**0.1091**	0.1089	0.1084
	10%	0.1177	0.1211	**0.1217**	0.1208	0.1205
	50%	0.1358	0.1372	0.1391	0.1398	**0.1404**
	100%	0.1406	0.1437	0.1447	0.1471	**0.1477**

Table 5: Effect of H on H@10

Scenarios	ϕ	H 0	1	2	3	4
Scenario 1	5%	0.2285	0.2323	**0.2354**	0.2261	0.2204
	10%	0.2329	0.2345	**0.2362**	0.2349	0.2338
	50%	0.2498	**0.2536**	0.2512	0.2509	0.2501
	100%	0.2549	**0.2562**	0.2547	0.2540	0.2534
Scenario 2	5%	0.0976	0.0996	0.1053	0.1094	**0.1143**
	10%	0.1066	0.1111	0.1131	**0.1152**	0.1127
	50%	0.1365	0.1377	0.1385	**0.1389**	0.1299
	100%	0.1372	**0.1426**	0.1422	0.1414	0.1365
Scenario 3	5%	0.1091	0.1097	0.1166	**0.1179**	0.1146
	10%	0.1217	0.1219	0.1260	**0.1286**	0.1250
	50%	0.1404	0.1408	**0.1426**	0.1406	0.1404
	100%	0.1477	**0.1481**	0.1469	0.1458	0.1440

DCDCSR [27] first computes combined user latent vectors by a convex combination of the latent vectors respectively obtained from the source-domain and target-domain, weighted by each user's rating sparsity. Then it learns a mapping function that captures the relationship from the target-domain latent vectors to the combined latent vectors. CoNet [11] transfers and combines the knowledge by using cross-connections between feed-forward neural networks, which are used for a base network for each domain. MINDTL [7] combines the CF information of the target-domain with the rating patterns extracted from a cluster-level rating matrix in the source-domain. Although all these methods are effective in improving the overall recommendation accuracy for target-domain users, they cannot make recommendations for cold-start users who do not interacted with any items.

CDR methods for cold-start users. Several approaches have been proposed to recommend target-domain items to cold-start users [5, 12, 18, 19]. CBMF [19] first factorizes a cluster-based cross-domain coarse matrix to learn the shared preferences within user and item clusters. Then it performs cross-domain recommendations by linearly combining the user-item interactions at the cluster-level and the user-level. EMCDR [18] proposes a general CDR framework. It learns a mapping function between the latent spaces of the source-domain and target-domain, by formulating the learning procedure as a supervised regression problem. On the other hand, due to the technical difficulties of the recommendation to cold-start users in the absence of CF information, most studies have focused on side information for user and item modeling [5, 6, 12]. MVDNN [4] and GCBAN [6] exploit users' demographic information (e.g. age, gender, and profession) or items' contextual information (e.g. genre of music, singer's name and publisher). RC-DFM [5] employs Additional Stacked Denoising Autoencoder (aSDAE) [3] to fuse such side information with CF information, then learns a mapping function by supervised regression. Our work is distinguished from the above methods in that we do not use any auxiliary data sources and also learn the cross-domain mapping based on a semi-supervised approach which is effective even in the case that only a small number of labeled data is available.

7 CONCLUSION

This paper proposes SSCDR, a CDR framework for cold-start users based on a semi-supervised approach. We first provide an analysis showing that only a few users overlaps between two domains in real-world CDR scenarios, and point out that the supervised approach is not effective due to the lack of labeled data. SSCDR models the users and items in the metric spaces where the similarities among users and items are represented as their distances, and trains a cross-domain mapping function using the distance-based loss defined by all the items (unlabeled data) as well as the overlapping users (labeled data). Furthermore, we introduce a multi-hop neighborhood inference technique to infer the latent vectors of cold-start users by fully utilizing their neighborhood information. Through our extensive experiments, we demonstrate that SSCDR learns the cross-domain relationship more accurately than the existing supervised approaches, which considerably improves the recommendation accuracy for the cold-start users.

ACKNOWLEDGEMENTS

This research was supported by the NRF grant funded by the MSIT: (No. 2016R1E1A1A01942642) and (No. 2017M3C4A7063570), the IITP grant funded by the MSIT: (No. 2018-0-00584) and (IITP-2019-2011-1-00783).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*.
[2] Hongxu Chen, Hongzhi Yin, Weiqing Wang, Hao Wang, Quoc Viet Hung Nguyen, and Xue Li. 2018. PME: Projected Metric Embedding on Heterogeneous Networks for Link Prediction. In *KDD*.
[3] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. 2017. A hybrid collaborative filtering model with deep structure for recommender systems. In *AAAI*.
[4] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*.
[5] Wenjing Fu, Zhaohui Peng, Senzhang Wang, yang xu, and Jin Li. 2019. Deeply Fusing Reviews and Contents for Cold Start Users in Cross-Domain Recommendation System. In *AAAI*.
[6] Jia He, Rui Liu, Fuzhen Zhuang, Fen Lin, Cheng Niu, and Qing He. 2018. A General Cross-Domain Recommendation Framework via Bayesian Neural Network. In *ICDM*.
[7] Ming He, Jiuling Zhang, Peng Yang, and Kaisheng Yao. 2018. Robust Transfer Learning for Cross-domain Collaborative Filtering Using Multiple Rating Patterns Approximation. In *WSDM*.
[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
[9] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*.
[10] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *WWW*.
[11] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *CIKM*.
[12] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. MTNet: A Neural Approach for Cross-Domain Recommendation with Unstructured Text. (2018).
[13] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *WWW*.
[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[15] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009).
[16] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer Learning for Collaborative Filtering via a Rating-matrix Generative Model. In *ICML*.
[17] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* (2008).
[18] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach.. In *IJCAI*.
[19] Nima Mirbakhsh and Charles X Ling. 2015. Improving top-n recommendation for cold-start users via cross-domain information. *TKDD* (2015).
[20] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *ICDM*.
[21] C. Park, D. Kim, X. Xie, and H. Yu. 2018. Collaborative Translational Metric Learning. In *ICDM*.
[22] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *JMLR* (2011).
[23] Parikshit Ram and Alexander G. Gray. 2012. Maximum Inner-product Search Using Cone Trees. In *KDD*.
[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*.
[25] Jason Weston, Frédéric Ratle, and Ronan Collobert. 2008. Deep Learning via Semi-supervised Embedding. In *ICML*.
[26] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*.
[27] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, Mehmet A Orgun, and Jia Wu. 2018. A Deep Framework for Cross-Domain and Cross-System Recommendations. In *IJCAI*.
[28] Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning* (2009).