RecSys-DAN: Discriminative Adversarial Networks for Cross-Domain Recommender Systems

Cheng Wang, *Member IEEE*, Mathias Niepert, Hui Li*

***Abstract*—Data sparsity and data imbalance are practical and challenging issues in cross-domain recommender systems. This paper addresses those problems by leveraging the concepts which derive from representation learning, adversarial learning and transfer learning (particularly, domain adaptation). Although various transfer learning methods have shown promising performance in this context, our proposed novel method RecSys-DAN focuses on alleviating the cross-domain and within-domain data sparsity and data imbalance and learns transferable latent representations for users, items and their interactions. Different from existing approaches, the proposed method transfers the latent representations from a source domain to a target domain in an adversarial way. The mapping functions in the target domain are learned by playing a min-max game with an adversarial loss, aiming to generate domain indistinguishable representations for a discriminator. Four neural architectural instances of ResSys-DAN are proposed and explored. Empirical results on real-world Amazon data show that, even without using labeled data (i.e., ratings) in the target domain, RecSys-DAN achieves competitive performance as compared to the state-of-the-art supervised methods. More importantly, RecSys-DAN is highly flexible to both unimodal and multimodal scenarios, and thus it is more robust to the cold-start recommendation which is difficult for previous methods.**

***Index Terms*— adversarial learning, neural networks, recommender systems, imbalanced data, domain adaptation**

I. Introduction

RECOMMENDER systems (RS) generate predictions based on the customers' preferences and purchasing histories. Collaborative filtering (CF) and content-based filtering (CBF) are popular techniques used in such systems [1]. CF-based methods generate recommendations by computing latent representations of users and products with matrix factorization (MF) methods [2]. Although CF-based approaches perform well in several application domains, they are based solely on the *sparse* user-item rating matrix and, therefore, suffer from the so-called *cold-start* problem [3]. For new users without a rating history and newly added products with few or no ratings (i.e., sparse historical data), the systems fail to generate high-quality personalized recommendations.

Alternatively, CBF approaches leverage auxiliary information such as product descriptions [4], locations [5] and social network [6] to generate recommendations. These methods are in principle more robust to cold-start problem as they can utilize different modalities. However, a pure CBF approach will face difficulties in learning sharable and transferable information of users and items across different product domains (e.g., "book" or "movie") [7]. A typical example of this scenario is cross-domain recommendation. Large online retailers such as Amazon and eBay often obtain user-item preferences from multiple domains so that the quality of recommendation could be improved by transferring knowledge acquired in a source domain to a target domain. The source-target data domain pairs in cross-domain recommendation are typically *imbalanced* in two aspects: *cross-domain imbalance* and *within-domain imbalance*. The former means that the numbers of users, items or labels in two domains are imbalanced (as shown in Tab. I), The latter refers to the problem that the distribution of categorical labels (i.e., rating scores) within one domain is imbalanced. Fig. 1 presents the imbalanced scenarios in 5-score based cross-domain recommendation. In this example, both cross-domain imbalance and within-domain imbalance exist.

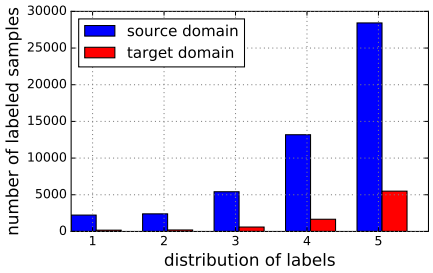


Fig. 1: The illustration of cross-domain imbalance and within domain imbalance problems in cross-domain recommendation problem. The *source domain* represents product domain "digital music" and *target domain* stands for product domain "music instrument" in Amazon dataset (see Section V-A for the detailed explanation of the dataset).

Alleviating the aforementioned data sparsity and data imbalance problems is a non-trivial issue for the cross-domain recommendation. However, existing CF-based and CBF-based approaches may fail to handle the problems when data becomes more and more sparse. One possible solution is to shift the learning schema from supervised to semi-supervised with limited labeled data. When it comes to a target domain in which the labeled data are completely unavailable, the only way to make a recommendation is transfer learning, particularly domain adaptation, by leveraging the knowledge from other domains.

To address the limitation of existing methods, in this paper,

*Corresponding author.

Cheng Wang is with the NEC Laboratories Europe, Heidelberg, Germany, e-mail: cheng.wang@neclab.eu.

Mathias Niepert is with the NEC Laboratories Europe, Heidelberg, Germany, e-mail: mathias.niepert@neclab.eu.

Hui Li is with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen, Fujian, P. R. China. e-mail: hui@xmu.edu.cn.

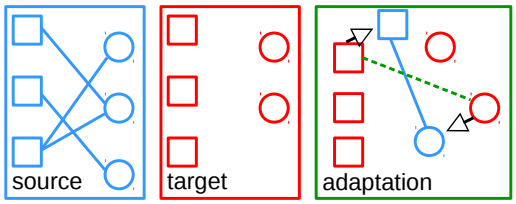


Fig. 2: Unsupervised adversarial adaptation for cross-domain recommendation. Each square presents a user and circle presents an item, the links between users and items present the preference information (rating) that users express on items. The rating scores are not available in the target domain. The dash links are generated by our proposed method with adversarial adaptation.

we propose a method called *Discriminative Adversarial Networks for Cross-Domain Recommendation (RecSys-DAN)* to learn the transferable latent representations of users, items and user-item pairs across different product domains. RecSys-DAN is rooted in the recent success of imbalanced learning [8], [9], [10], [11], [12], transfer learning [13] and adversarial learning [14], It adopts unsupervised adversarial loss function in combination with a discriminative objective.

A related research field to RecSys-DAN is domain adaptation [15]. Although domain adaptation has shown the capability to mitigate the rating sparsity problem, we argue that adversarial domain adaptation [16] for recommender systems has two distinct advantages. First, with unsupervised adversarial domain adaptation, we can learn a recommendation model when labels in the target domain are entirely not available, the typical domain adaption usually week or even not work in this case [17], [18], [19]. Second, we can observe the performance improvements that brought from adversarial domain adaptation as compared to traditional domain adaption, and we reported the evidence in Tab. II. Moreover, RecSys-DAN incorporates not only rating information but also additional user and item features such as product images and review texts. Fig. 2 demonstrates how RecSys-DAN aligns objects with different types and their existing preference relationships in order to predict new preference relationships in the target domain.

RecSys-DAN targets at the cold-start scenarios where no or only very few user-item preferences are available in the target domain. Existing supervised methods [20], [21], [22], [23], [24], [25], [26], [27] fail in this setting. We evaluate RecSys-DAN on real-world datasets and explore various scenarios where the information in the source and target domains are in the form of uni-modality or multi-modality. The experimental results show that RecSys-DAN achieves competitive performance compared to a variety of state-of-the-art supervised methods which have access to ratings in the target domain.

In summary, RecSys-DAN makes the following contributions:

- RecSys-DAN is the first neural framework adopting an adversarial loss for the cold-start problem that caused by data sparsity and imbalance in cross-domain recommender systems. It learns domain indistinguishable representations of different types of objects (users and items) and their interactions.
- RecSys-DAN is a highly flexible framework, which incorporates data in various modalities such as numerical, image and text.
- RecSys-DAN addresses the cross-domain data imbalance issue as well as imbalanced preferences in recommender systems by using representation learning and adversarial learning.
- RecSys-DAN achieves very competitive performance to the state-of-the-art supervised methods on real-world datasets where the target labels are completely not available.

The rest of this paper is organized as follows: Section II provides background and discusses related work. We present the motivation and problem statement in Section III. The details of our proposed approach, RecSys-DAN, are illustrated in Section IV. Experiments on real datasets that demonstrate the practicality and effectiveness of RecSys-DAN are presented in Section V. Section VI concludes our work.

II. Related Work

This work is related to four lines of work: cross-domain recommendation, imbalanced learning, adversarial learning and domain adaptation.

A. Cross-domain recommendation

Cross-domain recommendation (CDR) offers recommendations in a target domain by exploiting knowledge from source domains. To some extent, CDR can overcome the limitations of traditional recommendation approaches. It has been viewed as a potential solution to mitigate the cold-start and sparsity problem in recommender systems. Some methods have been proposed [28], [23] along this line. EMCDR [17] is proposed to learn a mapping function across domains. TCB [18] learns transferable contextual bandit policy for CDR. Sheng et al. [29] propose ONMTF, which is a non-negative matrix trifactorization based method. Xu et al. [19] recently propose a two-side cross-domain model (CTSIF_SVMs) which assumes that there are some objects (users and/or items) which can be shared in the user-side domain and item-side domain. Different to these methods, RecSys-DAN considers that target domain has completely unlabeled data (i.e., no ratings). Existing methods will encounter difficulties in learning effective models for such a scenario.

B. Imbalanced Learning

Recently, Imbalanced learning [8], [9], [10], [30] has been adapted to cross-domain data [11], [12]. Xue et. al [30] explore the theoretical explanations for re-balancing imbalanced data. Hsu et al. [11] propose a Closest Common Space Learning (CCSL) algorithm by exploiting both label and structural information for data within and across domains. This is achieved by learning data correlations [31] and related latent source-target domain pairs. RecSys-DAN is similar to CCSL, but it distinguishes itself by integrating representation learning and adversarial learning in recommender system domain. While

the typical cross-domain recommendation is in line with data imbalance problem, RecSys-DAN aims to transfer knowledge from a domain with abundant data to a domain with scarce data instead of directly re-balancing data.

C. Generative Adversarial Network (GANs)

Generative Adversarial Network (GANs) [14] is the most successful method in adversarial learning. Recently, many GAN-based extensions are proposed in different areas: image generation (e.g., DCGAN [32] and Wasserstein GAN [33]), NLP (e.g., SeqGAN [34]) and domain transfer problem [35]. In recommender systems community, IRGAN [36] is the first work to integrate GANs into item-based recommendation. Differently, RecSys-DAN can be viewed as the first work which explores the power of GAN in the context of cross-domain recommender systems.

D. Domain Adaptation

Transfer learning [13], [37] has been recently proposed to address the data sparsity problem in recommender systems [38], [39]. Domain adaptation, as a special form of transfer learning, arises with the hypothesis that large amounts of labeled data from a source domain are somehow similar to that in the unlabeled target domain. It has been applied to learn domain transferable representation in a variety of computer vision tasks [16], [40], [41], [35], [42]. Domain-Adversarial Neural Network (DANN) [16] learns domain-invariant features with adversarial training. Domain Transfer Network (DTN) [41] translates images across domains. E. Tzeng et al. propose a unified framework, Adversarial Discriminative Domain Adaptation (ADDA) [35], for object classification task. RecSys-DAN is partly inspired by ADDA, though there are many differences between ADDA and RecSys-DAN. RecSys-DAN is different to existing adaptation methods mainly in two aspects: RecSys-DAN adopts multi-level generators and discriminator for user/item features and their interactions, and it can captures features from multimodal data [43].

III. Motivation and Problem Statement

1) Motivation: Motivated by the success of GANs and domain adaptation, RecSys-DAN aims to address the data sparsity and data imbalance problem in a target domain by adapting the object (user or item) and their interactions from a source domain, i.e., learning to align user, item and user-item preference representations across domains via discriminative adversarial domain adaptation.

2) General Problem: We first formalize the typical setting of a recommender system. Let $\mathcal{D}$ be a dataset consisting of N users $U = \{\mathcal{U}_1, ..., \mathcal{U}_N\}$ and M items $V = \{\mathcal{V}_1, ..., \mathcal{V}_M\}$. The user-item preferences can be represented as a rating matrix $\mathcal{Y} \in \mathbb{R}^{N \times M}$, where $\mathcal{Y}_{uv}$ is user $\mathcal{U}$'s preference rating on item $\mathcal{V}$. We denote by $\mathcal{U} = V(\mathcal{U}) = \{\mathcal{V} \in V | \mathcal{Y}_{uv} \neq 0\}$, the set of items on which user $\mathcal{U}$ has non-zero preference values. Similarly, we use $\mathcal{V} = U(\mathcal{V}) = \{\mathcal{U} \in U | \mathcal{Y}_{uv} \neq 0\}$ to indicate the set of users who have non-zero ratings on item $\mathcal{V}$. The task of recommender systems is to learn a function h to predict the preference rating $\hat{\mathcal{Y}}_{uv}$ of user $\mathcal{U}$ for item $\mathcal{V}$ so that $\hat{\mathcal{Y}}_{uv}$ approximates ground-truth preference score $\mathcal{Y}_{uv}$. The function h often has the following form:

$$\hat{\mathcal{Y}}_{uv} = h(\mathcal{U}, \mathcal{V}; \Theta_h), \tag{1}$$

where Θ_h are the learnable parameters of h. The users and items are associated with existing features such as product metadata when available. The denser the user-item preference matrix $\mathbf{P}$ is, the less challenging the learning and prediction problems are. However, $\mathbf{P}$ can be very sparse in practice.

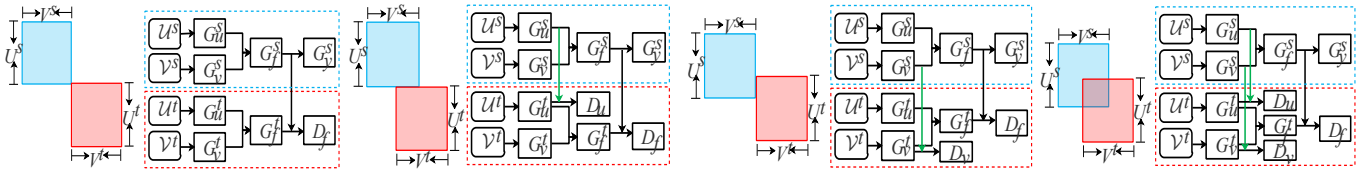
3) Adversarial Cross-Domain Alignment: To address this type of data sparsity problem, we propose to perform domain adaptation going from a *source* domain with several user-item preference values to a *target* domain with *no* user-item preferences. Specifically, the proposed approach learns a function G that maps the following objects to latent vector representations: the set of items that represented as $\mathcal{U}$; the set of users that represented as $\mathcal{V}$; the set of user-item pairs $(\mathcal{U}, \mathcal{V})$. The G is learned in a way that a discriminator D cannot distinguish the latent representations generated for the target domain from the latent representations generated for the source domain. We achieve this by introducing an adversarial learning loss involving G and D. For the sake of readability, we refer to G as a generator and write G_j^k to denote different types of generators with $k \in \{s, t\}$ (source or target) and $j \in \{u, v, f\}$ (user, item or item-user pairs).

Contrary to existing work, we formulate the adversarial loss for different types of objects (users and items) and their interactions. The adversarial loss, therefore, aligns distributions of latent items and user representations *as well as* their relationships given by the user-item preferences. The latent representations computed by the generators, therefore, fall into three categories: (1) user representations; (2) item representation; and (3) interaction representations of user-item pairs.

4) Shared Cross-Domain Objects: Learning across domains requires the existence of some relations in the participating domains. Usually, this relation is formed when objects (users, items) are found to be common in both domains [44]. To cover the different scenarios, RecSys-DAN includes four different adversarial cross- domain adaptation scenarios as below. They are classified according to whether a subset of user set U and item set V exists in both source and target domains:

- Interaction adaptation: $U^s \cap U^t = \varnothing$ and $V^s \cap V^t = \varnothing$.
- User adaptation: $U^s \cap U^t = \varnothing$ and $V^s \cap V^t \neq \varnothing$.
- Item adaptation: $U^s \cap U^t \neq \varnothing$ and $V^s \cap V^t = \varnothing$.
- Hybrid adaptation: $U^s \cap U^t \neq \varnothing$ and $V^s \cap V^t \neq \varnothing$.

Correspondingly, we proposed UI-DAN, U-DAN, I-DAN and H-DAN as shown in Fig. 3. The additional discriminators (in green) are introduced for shared objects. For instance, in the user adaptation scenario (U-DAN) where the set of users in the source and target domain are disjoint, we introduce a discriminator D_u attempting to distinguish between latent user representations from the source and target domain in order to align those representations in latent space.



(a) UI-DAN (b) U-DAN (c) I-DAN (d) H-DAN

Fig. 3: RecSys-DAN instantiations. U^k, V^k, $k \in \{s, t\}$ are user and item sets in domain k. The overlaps show that the shared user set of U^s and U^t, or shared item set of V^s and V^t . G_u, G_v, G_f (D_u, D_v, D_f) are corresponding to user, item and interaction feature generators (discriminators). The goal is: learning to align the latent representations between a source domain and a target domain that discriminators cannot distinguish. G_y^s is the scoring function in the source domain.

IV. Discriminative Adversarial Networks for Cross-Domain Recommendation

We firstly describe the learning of representations of objects (i.e., user and item) and their interactions. Then we elaborate on the objectives of learning to align the representations across domains. Finally, we introduce RecSys-DAN as a generalized adversarial adaptation framework.

A. Learning Domain Representations

Given a set of users, items and ratings in the source domain, we can learn a latent representation space $\mathcal{X}^s \in \mathbb{R}^d$ by computing a supervised loss on the given input X^s and ratings Y^s. Since the target domain has no or only few ratings, we do not directly learn the representations for the target domain. Instead, we learn mappings from the source representation space $\mathcal{X}^s$ to the target representation space $\mathcal{X}^t \in \mathbb{R}^d$ so as to minimize the distance between them. This can be achieved by first parameterizing source and target mapping functions, $M^s : X^s \to \mathcal{X}^s$ and $M^t : X^t \to \mathcal{X}^t$, and then minimizing the distance between the empirical source and target mapping distributions: $M^s(X^s)$ and $M^t(X^t)$ [35]. In this work, $M^k = \{G_u^k, G_v^k, G_f^k\}, k \in \{s, t\}$ is a set consisting of user mapping function G_u^k and item mapping function G_v^k, and user-item pair mapping function G_f^k.

For learning textual representations, the G_u^k is a recurrent neural network (RNN), specifically, RecSys-DAN adopts Long Short-Term Memory (LSTM) [45]:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
g_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{2}
$$

where i_t, f_t and o_t are input, forget and output gate respectively, c_t is memory cell. G_v^k can be either RNN-based (when review texts are used to represent an item) or convolutional neural network (CNN)-based for visual representations (when product image is used to represent an item). As shown in Fig. 3, the mapping function $G_u^k(\mathcal{U}^k; \Theta_u^k) : \mathcal{U}^k \to \mathcal{X}_u^k \in \mathbb{R}^d$ that maps a user sample to a d dimensional vector $\mathcal{X}_u^k$ and is parameterized by Θ_u^k. Similarly, we have the item mapping function $G_v^k(\mathcal{V}^k; \Theta_v^k) : \mathcal{V}^k \to \mathcal{X}_v^k \in \mathbb{R}^d$. Given user and item representations $(\mathcal{X}_u^k, \mathcal{X}_v^k)$, mapping function $G_f^k(\mathcal{X}_u^k, \mathcal{X}_v^k; \Theta_f^k) : (\mathcal{X}_u^k, \mathcal{X}_v^k) \to \mathcal{X}_f^k \in \mathbb{R}^d$ learns user-item interaction representation $\mathcal{X}_f^k$. The prediction $\hat{\mathcal{Y}} = h^k\Big(G_f^k(\mathcal{X}_u^k, \mathcal{X}_v^k; \Theta_f^k); \Theta_h^k\Big)$, where h^k is the scoring function in Eq. 1. Since there in only one scoring function can be learned in supervised way, i.e. G_y^s, and $h^s = h^t = G_y^s$, we use G_y^s to represent the scoring function. In a source domain, the parameters $\Theta^s = \{\Theta_u^s, \Theta_v^s, \Theta_f^s, \Theta_h^s\}$ are learned by optimizing the objective:

$$
\min_{\mathbf{\Theta^s}} \left[\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}^s(\mathcal{U}_i^s, \mathcal{V}_i^s, \mathcal{Y}_i^s) + \lambda \left\| \Theta^s \right\| \right] \tag{3}
$$

where $\langle \mathcal{U}_i^s,\ \mathcal{V}_i^s,\ \mathcal{Y}_i^s \rangle$ presents raw $\langle$user, item, truth score$\rangle$ triple, and $\mathcal{L}^s(\mathcal{U}_i^s, \mathcal{V}_i^s, \mathcal{Y}_i^s) = \| \hat{\mathcal{Y}}_i^s - \mathcal{Y}_i^s \|^2$. $|\mathcal{D}|$ is the size of training set. λ is the regularization parameter. By minimizing the objective function (3), the mapping functions G_u^s, G_v^s and G_f^s can be learned and used for extracting user, item and user-item features respectively in source domain by fixing corresponding parameter. For the unlabeled target domain, the corresponding target mapping functions G_u^t, G_v^t and G_f^t can be learned adversarially as we will explain in the next section.

B. Adversarial Representation Adaptation

One of the algorithmic principles of domain adaptation is to learn a space in which source and target domains are close to each other while keeping good performances on the source domain task [7]. Following the settings of standard GAN [14], domain discriminators D_u, D_v and D_f in RecSys-DAN are designed to perform min-max games and adversarially learn target generators (i.e., mapping functions) $G_u^t(\mathcal{U}^t; \Theta_u^t)$, $G_v^t(\mathcal{V}^t; \Theta_v^t)$ and $G_f^t(\mathcal{X}_u^t, \mathcal{X}_v^t; \Theta_f^t)$ with unlabeled samples. The loss functions of each instantiation of RecSys-DAN are as follows:

- UI-DAN: $\min\limits_{G_f^t} \max\limits_{D_f} \mathcal{L}(D_f, G_f^t)$
  $s.t.\ U^s \cap U^t = \varnothing$ and $V^s \cap V^t = \varnothing$.
- U-DAN: $\min\limits_{G_f^t, G_u^t} \max\limits_{D_f, D_u} \mathcal{L}(D_f, D_u, G_f^t, G_u^t)$
  $s.t.\ U^s \cap U^t = \varnothing$ and $V^s \cap V^t \neq \varnothing$
- I-DAN: $\min\limits_{G_f^t, G_v^t} \max\limits_{D_f, D_v} \mathcal{L}(D_f, D_v, G_f^t, G_v^t,)$
  $s.t.\ U^s \cap U^t \neq \varnothing$ and $V^s \cap V^t = \varnothing$

- H-DAN: $\min_{G_f^t, G_u^t, G_v^t} \max_{D_f, D_u, D_v} \mathcal{L}(D_f, D_u, D_v, G_f^t, G_u^t, G_v^t)$
$$s.t.\ U^s \cap U^t \neq \varnothing \text{ and } V^s \cap V^t \neq \varnothing$$

The objectives are learning generators in the target domain to generate features $\mathcal{X}^t \in \{\mathcal{X}_u^t, \mathcal{X}_v^t, \mathcal{X}_f^t\}$ which are intended to be close to the source latent representations $\mathcal{X}^s \in \{\mathcal{X}_u^s, \mathcal{X}_v^s, \mathcal{X}_f^s\}$. More specifically, G_f generates interaction-level domain indistinguishable features, while G_u/G_v generates indistinguishable user/item features for overlapping users/items. Formally, the source generators $M^s = \{G_f^s, G_u^s, G_v^s\}$ and predictor G_y^s is learned in a supervised way:

$$\begin{aligned}
&\min_{G_y^s, M^s} \mathcal{L}_s(U^s, V^s, Y^s) \\
&\quad = \mathbb{E}_{(\mathcal{U}^s, \mathcal{V}^s, \mathcal{Y}^s) \sim (U^s, V^s, Y^s)}[(G_y^s(M^s, \mathcal{U}^s, \mathcal{V}^s, \mathcal{Y}^s)] \\
&\quad = \frac{1}{|\mathcal{D}^s|} \sum_{i=1}^{|\mathcal{D}^s|} \mathcal{L}^s(\mathcal{U}_i^s, \mathcal{V}_i^s, \mathcal{Y}_i^s) + \lambda \left\|\Theta^s\right\| \\
&\quad = \frac{1}{|\mathcal{D}^s|} \sum_{i=1}^{|\mathcal{D}^s|} (\hat{\mathcal{Y}}_i^s - \mathcal{Y}_i^s)^2 + \lambda \left\|\Theta^s\right\|
\end{aligned} \tag{4}$$

The optimization of source weights Θ^s is formulated as a regression task which minimizes the mean squared error (MSE) over samples. In learning target generators $M^t = \{G_f^t, G_u^t, G_v^t\}$, M^s is used as a domain regularizer with fixed parameters. This is similar to the original GAN [14] where a generated space is updated with a fixed real space. To simplify, we take UI-DAN as an exemplary illustration, the learning objective is:

$$\begin{aligned}
&\max_{D_f} \mathcal{L}_f(U^s, V^s, U^t, V^t, M^s, M^t) \\
&\quad = \mathbb{E}_{(\mathcal{U}^s, \mathcal{V}^s) \sim (U_u^s, V_v^s)}[\log D_f(M^s(\mathcal{U}^s, \mathcal{V}^s))] \\
&\quad + \mathbb{E}_{(\mathcal{U}^t, \mathcal{V}^t) \sim (U_u^t, V_v^t)}[\log(1 - D_f(M^t(\mathcal{U}^t, \mathcal{V}^t)))]
\end{aligned} \tag{5}$$

$$\begin{aligned}
&\min_{M^t} \mathcal{L}_m(U^t, V^t, D_f) \\
&\quad = \mathbb{E}_{(\mathcal{U}^t, \mathcal{V}^t) \sim (U^t, V^t)}[\log(1 - D_f(M^t(\mathcal{U}^t, \mathcal{V}^t)))]
\end{aligned} \tag{6}$$

where M^t is initialized with M^s.

With learned M^t, user, item, interaction representations $\mathcal{X}_u^t$, $\mathcal{X}_v^t$, $\mathcal{X}_f^t$ can be extracted as inputs for scoring function G_y^s, which makes preference predictions. Note that one of the essential differences between RecSys-DAN and prior recommendation methods is that ResSys-DAN takes the cross-domain overlap users (items) into account to learn indistinguishable user (item) representation as shown in Fig. 3b, Fig. 3c, and Fig. 3d. With shared users and items across domain, additionally, D_u and D_v are designed and lead to scenarios that have interaction-level D_f, G_f and feature-level D_u, D_v, G_u, G_v:

$$\begin{aligned}
&\max_{D_z} \mathcal{L}_z(U^s, V^s, U^t, V^t, G_u^s, G_v^s, G_u^t, G_v^t) \\
&\qquad\qquad \min_{G_z^t} \mathcal{L}_m(U^t, V^t, D_z) \\
s.t.\ \ D_z = D_u &\ \text{ if } U^s \cap U^t = \varnothing \text{ and } V^s \cap V^t \neq \varnothing \\
D_z = D_v &\ \text{ if } U^s \cap U^t \neq \varnothing \text{ and } V^s \cap V^t = \varnothing \\
D_z = D_u, D_v &\ \text{ if } U^s \cap U^t \neq \varnothing \text{ and } V^s \cap V^t \neq \varnothing
\end{aligned} \tag{7}$$

The optimization of the additional discriminators and generators is achieved by fine-tuning G_u^t (G_v^t) on cross-domain shared user/item subset.

Algorithm 1: Learning algorithm for UI-DAN

Input: source set $\mathcal{D}^s = \{X_u^s, X_v^s, Y^s\}$, target set $\mathcal{D}^t = \{X_u^t, X_v^t\}$, dummy domain label $Y^d \in \{0, 1\}$, batch size $\mathcal{B}$.
Initilize: M^s, M^t, G_y^s, D_f
$\mathcal{N}^s = |\mathcal{D}^s|$, $\mathcal{N}^t = |\mathcal{D}^t|$
pre-train on source domain:
repeat
  for $b \leq \frac{\mathcal{N}^s}{\mathcal{B}}$ **do**
    mini batch $(\mathcal{U}_b^s, \mathcal{V}_b^s, \mathcal{Y}_b^s) \in (X_u^s, X_v^s, Y^s)$
    $M^s, G_y^s \Leftarrow \min \mathcal{L}_s(\mathcal{U}_b^s, \mathcal{V}_b^s, \mathcal{Y}_b^s)$
until *stopping criterion is met*;
train generators on target domain:
set $M^t \Leftarrow M^s$, and fix M^s
repeat
  for $b \leq \frac{\mathcal{N}^s}{\mathcal{B}}$ **do**
    mini batch $(\mathcal{U}_b^s, \mathcal{V}_b^s) \in (X_u^s, X_v^s)$
    for $k \leq \frac{\mathcal{N}^t}{\mathcal{B}}$ **do**
      mini batch $(\mathcal{U}_k^t, \mathcal{V}_k^t) \in (X_u^t, X_v^t)$
      $D_f \Leftarrow \max \mathcal{L}_f(\mathcal{U}_b^s, \mathcal{V}_b^s, \mathcal{U}_k^t, \mathcal{V}_k^t, \mathcal{Y}^d)$
      $M_t \Leftarrow \min \mathcal{L}_m(\mathcal{U}_k^t, \mathcal{V}_k^t)$
until *stopping criterion is met*;
Output: M^t
inference on target domain:
$\hat{y}_t \Leftarrow G_y^s(M^t(x_u^t, x_v^t))$

C. Generalized Framework

RecSys-DAN is a generalized framework. The choice of RecSys-DAN instantiations is based on considering the following questions: (1) Which type of modalities (e.g. numerical rating, review or image) are used to represent $\mathcal{U}$ and $\mathcal{V}$? (2) Are there shared users and/or items across domains? (3) Which adversarial objective is used?

The training procedure of each instantiation is different to each other, but they also share some similarities. Algorithm 1 summaries the learning procedure of UI-DAN in which two training stages are involved. First, the pre-training in the source domain for obtaining source generators M^s and scoring function G_y^s. The update of parameters $\Theta_u^s, \Theta_v^s, \Theta_f^s$ are achieved by:

$$\Theta_j^s := \Theta_j^s + \eta \nabla_{\Theta_j^s} \frac{1}{\mathcal{B}} \sum_{i=1}^{\mathcal{B}} \mathcal{L}_s(\mathcal{U}_i^s, \mathcal{V}_i^s, \mathcal{Y}_i^s), \quad j \in \{u, v, f\} \tag{8}$$

where $\mathcal{B}$ is a min-batch of training samples, η is learning rate. Similarly, the optimal weights for scoring function $G_y^s(G_f^s; \Theta_y^s)$ can be learned. Second, cross-domain adversarial learning, the goal is to learn the target generators M^t in an adversarial way. By using dummy domain labels, $y^d = 1$ presents the data from source domain and $y^d = 0$ for target domain. The domain discriminator $D_f(G_f^s, G_f^t; \Theta_d)$ is obtained by ascending stochastic gradients [14] at each batch using the following update rule:

$$\Theta_d := \Theta_d + \eta \nabla_{\Theta_d} \frac{1}{\mathcal{B}} \sum_{i=1}^{\mathcal{B}} \mathcal{L}_f(\mathcal{U}_i^s, \mathcal{V}_i^s, \mathcal{U}_i^t, \mathcal{V}_i^t, \mathcal{Y}_i^d) \tag{9}$$

Note that target generators M^t is initialized with and updated in similar way as M^s. By doing this, M^t tries to push the user-item interaction representations in the target domain as close as possible to the source domain. Additionally, the ratings (i.e., labels) in the target domain are never accessed in learning procedures of RecSys-DAN. As a comparison, existing recommendation methods fail to handle this scenario. With learned M^t, the rating regression can be performed with source score function G_y^s for a given user-item pair in the target domain:

$$\hat{\mathcal{Y}}_{uv} \Leftarrow G_y^s\big(M^t(\mathcal{U}^t, \mathcal{V}^t)\big). \tag{10}$$

The learning procedures of U-DAN, I-DAN and H-DAN have additional fine-tuning stage with training samples of shared users/items. Algorithm 2 presents the learning for U-DAN and I-DAN while H-DAN is a combination of them.

Algorithm 2: Learning for U-DAN and I-DAN

Input: $\mathcal{D}^s = \{X_u^s, X_v^s, Y^s\}$, $\mathcal{D}^t = \{X_u^t, X_v^t\}$,
shared item set $\mathcal{D}_u^o = \{X_v^o, X_u^s, X_u^t\}$,
shared user set $\mathcal{D}_v^o = \{X_u^o, X_v^s, X_v^t\}$, $Y^d \in \{0, 1\}$.
Initialize: $M^s, M^t, G_y^s, D_u, D_v, D_f$
call Algorithm 1 to obtain M^t, learning rate $\eta \times 0.001$
learning U-DAN:
repeat
  for *each batch b,* $(\mathcal{V}_b^o, \mathcal{U}_b^s, \mathcal{U}_b^t) \in (X_v^o, X_u^s, X_u^t)$ **do**
    $D_u^t \Leftarrow \max \mathcal{L}_f(\mathcal{V}_b^o, \mathcal{U}_b^s, \mathcal{U}_b^t, \mathcal{Y}_b^d)$
    $G_u^t \Leftarrow \min \mathcal{L}_m(\mathcal{V}_b^o, \mathcal{U}_b^t)$
until *stopping criterion is met*;
learning I-DAN:
repeat
  for *each batch b,* $(\mathcal{U}_b^o, \mathcal{V}_b^s, \mathcal{V}_b^t) \in (X_u^o, X_v^s, X_v^t)$ **do**
    $D_v^t \Leftarrow \max \mathcal{L}_f(\mathcal{U}_b^o, \mathcal{V}_b^s, \mathcal{V}_b^t, \mathcal{Y}_b^d)$
    $G_v^t \Leftarrow \min \mathcal{L}_m(\mathcal{U}_b^o, \mathcal{V}_b^t)$
until *stopping criterion is met*;
Output: G_u^t, G_v^t

V. Experiments

This section evaluates the performance of RecSys-DAN on both unimodal and multimodal scenarios.

A. Dataset and Evaluation Metric

We evaluated RecSys-DAN on multiple sets on the Amazon dataset [46][1], which is widely used for evaluating recommender systems [22], [27]. It contains different item and user modalities such as review text, product images and ratings. We selected 5 categories to form three (source → target) domain pairs: Digital Music→Music Instruments (DM→MI), Home & Kitchen→Office Products (HK→OP) and CDs & Vinyl→Digital Music (CDs→DM). Some statistics of the datasets are listed in Tab. I. $|\mathcal{VOC}|$ is the size of the vocabulary of words used in reviews in the source and target training sets. Words which occurred less than 5 times were removed. We randomly split each dataset into 80%/10%/10% for training/validation/test. The training reviews associated with a user/item were concatenated to present the user/item following previous work [27]. We aligned users (items) that occured in both the source and target domains to ensure an equal number of training reviews for both domains. We evaluated all the models on the rating prediction task using both the root mean squared error (RMSE) and the mean average error (MAE):

TABLE I: Overview of the datasets († presents training samples for shared users and items respectively)

$D^s \rightarrow D^t$	User	Item	Sample	$\lvert\mathcal{VOC}\rvert$
DM	5540	3558	64544	4696
MI	1429	891	10156	
DM ∩ MI	23	0	23	
HK	14285	3227	41810	3651
OP	4773	1312	28044	
HK ∩ OP	1709	0	1709	
CDs	41437	9650	84432	10355
DM	5540	3558	6615	
CDs ∩ DM	4394	829	19529/6216†	

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{D}|}\sum_{(\mathcal{U},\mathcal{V})\in\mathcal{D}}(\hat{\mathcal{Y}}_{uv} - \mathcal{Y}_{uv})^2}, \quad \text{MAE} = \frac{1}{|\mathcal{D}|}\sum_{(\mathcal{U},\mathcal{V})\in\mathcal{D}}|\hat{\mathcal{Y}}_{uv} - \mathcal{Y}_{uv}| \tag{11}$$

where $\hat{\mathcal{Y}}_{uv}$ and $\mathcal{Y}_{uv}$ are predicted and truth rating, respectively.

B. Baseline Methods

We compare RecSys-DAN against a variety of methods. Naive: **Normal** is a random rating predictor which gives predictions based on the (norm) distribution of the training set. Matrix factorization: **NMF** [20], Non-negative Matrix Factorization that only uses ratings. And **SVD++** [21], extended SVD for latent factor modeling. Nearest neighbors: **KNN** [47]. Topic modeling: **HFT** [22]. Deep learning methods: **DeepCoNN** [27], which is the current state-of-the-art approach. Additionally, we compared RecSys-DAN with typical cross-domain recommendation methods.

Following previous work [40], [35], **source-only** results for applying a source domain models to the target domain are also reported. Note that rating information in the target domain is accessible to the baseline methods (except source-only), while RecSys-DAN has no access to ratings in the target domain.

C. Implementations

We implemented RecSys-DAN with Theano[2]. The discriminators D_f, D_u, D_v are formed with following layers: Dense(512)→Relu(·)→Dense(2)→Softmax(·). The architecture of generators varies according to different scenarios. For unimodal scenario (textual user and item representations), G_u^s, G_v^s, G_u^t, G_v^t are formed by: Embedding($|\mathcal{VOC}|$)→LSTM (256)→Average Pooling, and G_f^s, G_f^t are constructed using: Dense(512)→Dropout (0.5). For multimodal scenario (textual user representation and visual item representation), the main architecture of G_v^s, G_v^t is: CNN→Dense (4096)→ Dense(256), and other configurations remain unchanged as in unimodal scenario. The weights of LSTM are orthogonally initialized [48]. We used a batch size of 512. The models were optimized with ADADELTA [49] and the initial learning rate η is

[1]http://jmcauley.ucsd.edu/data/amazon/

[2]http://www.deeplearning.net/software/theano/

TABLE II: The results for UI-DAN and I-DAN in the unimodal and multimodal settings (s: source-only, a: adaptation, u: unimodal, m: multimodal). The best (supervised) baselines are in **blue**, and the best unimodal (multimodal) results of RecSys-DAN are in **green** (**red**) . $\Delta = (2\,|S^*_- - S^*_+|)/(S^*_- + S^*_+)$ presents the percentage differences between the best result of ours (S^*_-, in green) and that of baselines (S^*_+, in blue). It demonstrates how close the performance of (unsupervised) RecSys-DAN to the performance of (supervised) baselines.

$\mathcal{D}^s \rightarrow \mathcal{D}^t$	DM→MI		HK → OP		Target Domain Training Data		
Models	RMSE	MAE	RMSE	MAE	Rating	Review	Image
Normal	1.165 ± 0.022	0.843 ± 0.025	1.194 ± 0.024	0.894 ± 0.023	Yes	No	No
KNN	1.040 ± 0.000	0.709 ± 0.000	0.957 ± 0.000	0.710 ± 0.000	Yes	No	No
NMF	0.922 ± 0.009	0.644 ± 0.007	0.866 ± 0.003	0.637 ± 0.005	Yes	No	No
SVD++	0.891 ± 0.008	0.648 ± 0.006	**0.844 ± 0.002**	0.642 ± 0.002	Yes	No	No
HFT	0.914 ± 0.000	0.704 ± 0.000	0.917 ± 0.000	0.735 ± 0.000	Yes	Yes	No
DeepCoNN	**0.868 ± 0.002**	**0.599 ± 0.003**	0.875 ± 0.001	**0.634 ± 0.001**	Yes	Yes	No
UI-DAN (s, u)	1.087± 0.180	0.918± 0.002	0.959± 0.028	0.684± 0.003	No	Yes	No
I-DAN (s, u)	1.052± 0.220	0.884± 0.264	0.957± 0.033	0.684± 0.002	No	Yes	No
UI-DAN (s, m)	1.043± 0.056	0.879± 0.089	1.037± 0.008	0.875± 0.011	No	Yes	Yes
I-DAN (s, m)	1.450± 0.291	1.296± 0.308	1.953± 0.290	1.759± 0.286	No	Yes	Yes
UI-DAN (a, u)	0.920± 0.223	**0.674± 0.021**	0.917± 0.005	0.674± 0.002	No	Yes	No
I-DAN (a, u)	**0.914± 0.002**	0.675± 0.021	**0.911± 0.002**	**0.670± 0.002**	No	Yes	No
UI-DAN (a, m)	**0.991± 0.077**	**0.765± 0.143**	**0.934± 0.004**	**0.745± 0.006**	No	Yes	Yes
I-DAN (a, m)	1.078± 0.033	0.795± 0.027	1.144± 0.078	0.868± 0.039	No	Yes	Yes
Δ	5.16%± 0.22%	11.78%± 1.88%	7.64%±0.23%	5.52% ±0.23%	-	-	-

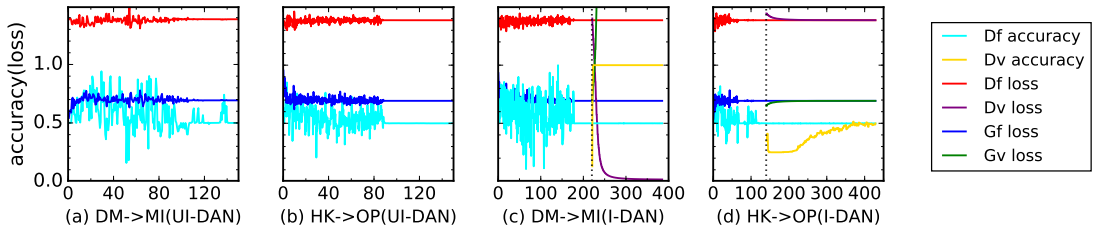


Fig. 4: Learning unimodal UI-DAN and I-DAN. It plots the changes of loss and accuracy of interaction-level (a-b) discriminator D_f/generator G_f and item-level discriminator D_v/generator G_v (c-d) on two dataset pairs against training epochs. The dash vertical lines in (c-d) denote the starting point for fine-tuning I-DAN. The X-axis presents the number of training epochs.

0.0001 (decreased by $\times$0.001 for U-DAN, I-DAN and H-DAN). We implemented KNN, NMF and SVD++ using SurPrise package[3] and used authors' implementations for HFT[4] and DeepCoNN[5]. To make a fair comparison, implemented baselines are trained with grid search (for NMF and SVD++, regularization [0.0001, 0.0005, 0.001], learning rate [0.0005, 0.001, 0.005, 0.01]. For HFT, regularization [0.0001, 0.001, 0.01, 0.1, 1], lambda [0.1, 0.25, 0.5, 1]). For DeepCoNN, we use the suggested default parameters. The best scores are reported.

D. Results and Discussions

We first evaluated two RecSys-DAN instances: UI-DAN (applied to the scenario where source and target domains have neither overlapping users nor items) and I-DAN (applied to the scenario where the source and target domains only shared some users) in the unimodal and multimodal scenarios. The results are summarized in Tab. II.

1) Unimodal RecSys-DAN: The results listed in Tab. II show that both UI-DAN and I-DAN improve the source-only baselines. For instance, UI-DAN reduces the source-only error by ∼15% (RMSE) and ∼27% (MAE) on DM→MI. On HK→OP, it improves the source-only baselines by ∼4% (RMSE) and ∼1.5% (MAE), respectively. In the scenario where source and target domains share users, I-DAN can improve UI-DAN on both dataset pairs (∼0.4% on average across metrics). Compared to its source-only baselines, I-DAN achieves improvements similar to those of UI-DAN.

Fig. 4a and Fig. 4b show the changes of the loss/accuracy of the interaction discriminator D_f and the loss of G_f against the number of epochs with the UI-DAN. On both dataset pairs, the equilibrium points are reached at ∼100 epochs where binary classification accuracy of discriminator is 50%. It suggests that the user-item interaction representation from generator is indistinguishable to discriminator. When training I-DAN with shared user samples, we first trained interaction-level D_f and G_f and then fine-tuned item-level D_v and G_v by decreasing learning rate to $0.001\times\eta$. We adopted small learning rate η to ensure that G_v could generate indistinguishable item representation for shared users while maintaining interaction-level representations. Figures 4c and 4d present the training procedure of I-DAN. On DM→MI, D_v and G_v had difficulty to converge due to limited shared user samples. On the contrary, with more shared samples, I-DAN was able to converge on both interaction-level and item-level on HK→OP. From experimental results, we can observe that item-level representations are not as important as interaction-level representation on rating prediction task. Similar findings are reported in Tab. III.

2) Multimodal RecSys-DAN: The task becomes more challenging when both ratings and reviews are not available. In

[3]http://surpriselib.com/
[4]http://cseweb.ucsd.edu/~jmcauley/code/code_RecSys13.tar.gz
[5]https://github.com/chenchongthu/DeepCoNN

TABLE III: RecSys-DAN Results on CDs →DM

$\mathcal{D}^s \rightarrow \mathcal{D}^t$ Models	CDs→DM RMSE	CDs→DM MAE
Normal	1.452 ± 0.021	1.100 ± 0.022
KNN	1.110 ± 0.000	0.870 ± 0.000
NMF	1.062 ± 0.001	0.861 ± 0.001
SVD++	1.061 ± 0.000	0.841 ± 0.001
HFT	1.099 ± 0.000	0.869 ± 0.000
DeepCoNN	**1.038 ± 0.004**	**0.805 ± 0.003**
Source Only	1.131± 0.028	0.857± 0.080
UI-DAN	1.076± 0.002	0.791± 0.019
U-DAN	1.071± 0.005	0.784± 0.002
I-DAN	1.068± 0.006	0.781± 0.002
H-DAN	**1.068± 0.002**	**0.779± 0.002**
Δ	2.85%± 0.28%	**3.28%± 0.32%**

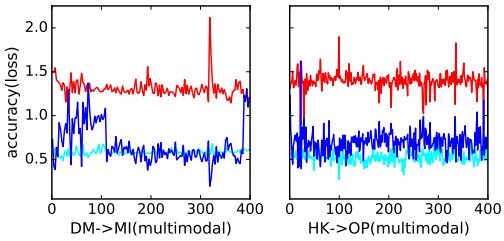


Fig. 5: Learning multimodal UI-DAN. The labels and legends are the same as Fig. 4.

this scenario, we replaced the review text of an item with its image, if available, which leads to a multimodal unsupervised adaptation problem. The correlations between textual user embeddings and visual item embeddings need to be adapted across the given domains. The results of UI-DAN (a, m) and I-DAN (a, m) in the multimodal settings can be found in Tab. II. We find that it is more difficult to learn user-item correlations across modalities, compared to the unimodal setting. Fig. 5 presents the learning of multimodal adversarial adaptation paradigm. Although the performance of multimodal UI-DAN and I-DAN is not as good as the unimodal ones, it is still robust when addressing the item-based cold-start recommendation problem. UI-DAN (a, m) and I-DAN (a, m), however, significantly improve UI-DAN (s, m) and I-DAN (s, m). For instance, I-DAN (a, m) outperforms I-DAN (s, m) by ∼26% (RMSE)/∼39% (MAE) for DM→MI and ∼41% (RMSE)/∼51% (MAE) for HK→OP, respectively.

3) Compare Different Instances of RecSys-DAN: An experiment was conducted on CDs→DM (unimodal) where both shared users and items existed to further explore the different instances of RecSys-DAN. The results in Tab. III illustrate that unsupervised domain adaptation models improve source-only baseline by ∼4.8% (RMSE) and ∼7.7% (MAE). We find that U-DAN, I-DAN and H-DAN did not bring significant improvements over UI-DAN. This is similar to the results of I-DAN and UI-DAN in Tab. II. We conjecture the main reason is that the rating prediction task is primarily based on the user-item interactions (e.g., users express preferences on items). The interaction representations are therefore of crucial importance as compared to user-level and item-level representations, though the shared users/items could be beneficial when connecting domains.

4) Compare to Cross-domain Recommendation Models: We now compare our proposed architectures with the state-of-the-art supervised models. As the first attempt to utilize unsupervised adversarial domain adaptation for the (cold-start) cross-domain recommendation, it is difficult to directly compare RecSys-DAN with previous methods. Existing cross-domain (e.g., EMCDR [17], CrossMF [23], HST [25]) or hybrid collaborative filtering (e.g., DeepCoNN [27], cmLDA [50]) methods are *NOT* able to learn models in the scenarios where ratings and/or review texts are completely not available for training. The Tab. IV suggests previous methods' limitations, which are addressed by our proposed adversarial domain adaptation method. Therefore, we compare RecSys-DAN with supervised baselines indirectly.

TABLE IV: The comparison with RecSys-DAN and existing cross-domain recommendation methods. Existing methods have difficulties in learning a recommendation model when ratings on the target domain are completely missing.

Methods	Required Target Inputs	Target Learning
EMCDR [17]	rating	supervised
DeepCoNN [27]	rating, review	supervised
DLSCF [51]	rating, binary rating	supervised
CrossMF [23]	rating	supervised
CTSIF_SVMs [19]	rating	supervised
HST [25]	ratings	supervised
cmLDA [50]	rating, review, description	supervised
RecSys-DAN	review or image	adversarial

5) Compare to Supervised Models: We trained the baselines directly on the target domain with labeled samples (Normal, KNN, NMF and SVD++ were trained with user-item ratings, while HFT and DeepCoNN were trained with both ratings and reviews). The goal is to examine how close the performance of unsupervised RecSys-DAN without labeled target data to those supervised methods which can access labeled target data. The results are reported in Tab. II and Tab. III. By purely transferring the representations learned in the source domain to the target domain, our methods achieve competitive performance compared to strong baselines. Specifically, RecSys-DAN is able to achieve similar performance as NMF and SVD++ with unsupervised adversarial adaptation and it outperforms baselines on MAE in Tab. III. From the aforementioned analysis, we can conclude that ResSys-DAN has much better generalization ability and it is more suitable to address practical problems such as cold-start recommendation.

6) Representation Alignment: To examine the extent to which the adversarial objective aligns the source and target latent representations, we randomly selected 2,000 test samples (1,000 from the source and 1,000 from the target domain) for extracting latent representations with G_f at different epochs. Fig. 6 visualizes the source and target domain representations. The source domain models' parameters are not updated during the adversarial training of the target generators. Comparing the representations at the 0^{th} epoch (no adaptation) and 50^{th}, 100^{th}, 200^{th} epochs, we can find that the distance between the latent representations of the source and target domains is decreasing during adversarial learning, making target representations more indistinguishable to source representations. Fig. 7 shows the visualization of weights for source and target domains after training. We can observe that the weights of the target mapping function G_f^t approximate those of source mapping function G_f^s, which again demonstrates that RecSys-DAN succeeds in

TABLE V: Exemplary predictions of RecSys-DAN (UI-DAN) on the target test set of "office product" with HK→OP cross-domain recommendation. The first two examples are unimodal and the last two examples are multimodal based prediction. The predictions are purely based on transferring the representations of user-item interaction in the source domain ("home & kitchen") via an unsupervised and adversarial way. "<UNK>" means the word is not included in built vocabulary dictionary $\mathcal{VOC}$. We removed punctuations in reviews.

Reviews written by user	Reviews and/or Images associated to item	Prediction	Truth
has four internal pockets which is a nice addition round rings but with the better <UNK> closure handy but could use slight improvement (...)	just what we needed good item great organizer less useful than i thought although may be just right for some colorful organizing okay (...)	**4.58**	**5**
worked well very cool great product great product works great awesome product well very easy to use	need a computer excellent for keeping organized in class durable easy to use super nice for presentations great quality and price great idea to (...)	**5.08**	**5**
good tape <UNK> not very good flow good boxes but they come <UNK>	 efficient tool best value for price while it lasted no frills sturdy sharpener for frequent pencil <UNK> sharp works as it should noisy but good excellent maybe not perfect for your use (...)	**4.15**	**4**
make sure you are on 24 <UNK> wifi nice little printer must have unit cost too high nice <UNK>	 great a really nice little remote what a treat for powerpoint presentations only on some <UNK> simple perfection (...)	**4.67**	**5**

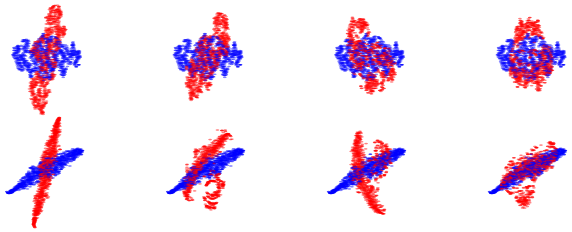


Fig. 6: t-SNE [52] visualizations of source (blue) and target (red) domain representations from UI-DAN (DM→MI (top), HK→OP (bottom)) at the 0th (no adaptation), the interaction representation adaptation at the 50th, 100th and 200th epochs.

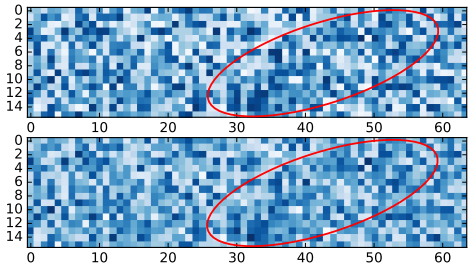


Fig. 7: The visualization of weights $\Theta_f^s, \Theta_f^t \in \mathbb{R}^{512\times512}$ for source (top) and target (bottom) domains. The model is trained with DM→MI domain pair, we only show the top-left 16×64 of weight matrix for readability. The red circles highlight the patterns that shared by the weights of source and target domains.

aligning the representations of the source and target domains through adversarial learning.

7) Cold-Start Recommendation: Tab. V presents some random rating prediction examples with pre-trained RecSys-DAN models in unimodal and multimodal scenarios. We can observe that representing users and items with reviews can effectively alleviates the cold-start recommendation problem when ratings are completely not available, since the proposed adversarial adaptation transfers the user, item and their interaction representations from a labeled source domain to an unlabeled target domain. It demonstrates the superiority of RecSys-DAN in making preference prediction without the access to label information (i.e., ratings in this example). The existing recommendation methods [20], [21], [47], [22], [27] fail in this scenario.

8) Running Time: The pre-training of RecSys-DAN in the source domain took ∼10 epochs (avg. 69s/epoch). The adversarial training in both source and target domains took ∼100 epochs to reach an equilibrium point. For inference, our model performs as fast as baseline models, since RecSys-DAN directly adapts the source scoring function.

VI. Conclusion

RecSys-DAN is a novel framework for cross-domain collaborative filtering, particularly, the real-world cold-start recommendation problem. It learns to adapt the user, item and user-item interaction representations from a source domain to a target domain in an unsupervised and adversarial fashion. Multiple generators and discriminators are designed to adversarially learn target generators for generating domain-invariant representations. Four RecSys-DAN instances, namely, UI-DAN, U-DAN, I-DAN, and H-DAN, are explored by considering different scenarios characterized by the overlap of users and items in both unimodal and multimodal settings. Experimental results demonstrates that RecSys-DAN has a competitive performance compared to state-of-the-art supervised methods for the rating prediction task, even with absent preference information.

References

[1] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[2] Hui Li, Tsz Nam Chan, Man Lung Yiu, and Nikos Mamoulis. FEXIPRO: fast and exact inner product retrieval in recommender systems. In *SIGMOD Conference*, pages 835–850, 2017.

[3] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260, 2002.

[4] Cheng Wang, Mathias Niepert, and Hui Li. LRMM: learning to recommend with missing modalities. In *EMNLP*, pages 3360–3370, 2018.

[5] Ziyu Lu, Hui Li, Nikos Mamoulis, and David W Cheung. Hbgg: a hierarchical bayesian geographical model for group recommendation. In *SDM*, 2017.

[6] Hui Li, Dingming Wu, Wenbin Tang, and Nikos Mamoulis. Overlapping community regularization for rating prediction in social recommender systems. In *RecSys*, pages 27–34, 2015.

[7] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[8] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In *IJCNN*, pages 1322–1328, 2008.

[9] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.

[10] Haibo He and Yunqian Ma. *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons, 2013.

[11] Tzu-Ming Harry Hsu, Wei-Yu Chen, Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with imbalanced cross-domain data. In *ICCV*, pages 4121–4129, 2015.

[12] Yao-Hung Hubert Tsai, Cheng-An Hou, Wei-Yu Chen, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Domain-constraint transfer coding for imbalanced unsupervised domain adaptation. In *AAAI*, pages 3597–3603, 2016.

[13] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *ICML*, pages 617–624, 2009.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[15] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*, pages 1041–1048, 2009.

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[17] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: an embedding and mapping approach. In *IJCAI*, pages 2464–2470, 2017.

[18] Bo Liu, Ying Wei, Yu Zhang, Zhixian Yan, and Qiang Yang. Transferable contextual bandit for cross-domain recommendation. *AAAI*, 2018.

[19] Xu Yu, Yan Chu, Feng Jiang, Ying Guo, and Dunwei Gong. Svms classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features. *Knowledge-Based Systems*, 141:80–91, 2018.

[20] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[21] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

[22] Julian J. McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, pages 165–172, 2013.

[23] Tomoharu Iwata and Takeuchi Koh. Cross-domain recommendation without shared users or items by sharing latent vector distributions. In *AISTATS*, pages 379–387, 2015.

[24] Chung-Yi Li and Shou-De Lin. Matching users and items across domains to improve the recommendation quality. In *KDD*, pages 801–810, 2014.

[25] Yan-Fu Liu, Cheng-Yu Hsu, and Shan-Hung Wu. Non-linear cross-domain collaborative filtering via hyper-structure transfer. In *ICML*, pages 1190–1198, 2015.

[26] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.

[27] Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*, pages 425–434, 2017.

[28] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *KDD*, pages 1285–1293, 2012.

[29] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, Zhanyu Ma, and Jun Guo. A cross-domain recommendation model for cyber-physical systems. *IEEE Trans. Emerging Topics Comput.*, 1(2):384–393, 2013.

[30] Jing-Hao Xue and Peter Hall. Why does rebalancing class-unbalanced data improve AUC for linear discriminant analysis? *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(5):1109–1112, 2015.

[31] Zhanyu Ma, Jing-Hao Xue, Arne Leijon, Zheng-Hua Tan, Zhen Yang, and Jun Guo. Decorrelation of neutral vector variables: Theory and applications. *IEEE Trans. Neural Netw. Learning Syst.*, 29(1):129–143, 2018.

[32] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[33] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

[34] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[35] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017.

[36] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR*, pages 515–524, 2017.

[37] Weike Pan and Qiang Yang. Transfer learning in heterogeneous collaborative filtering domains. *Artificial intelligence*, 197:39–55, 2013.

[38] Weike Pan. A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing*, 177:447–453, 2016.

[39] Lili Zhao, Sinno Jialin Pan, Evan Wei Xiang, Erheng Zhong, Zhongqi Lu, and Qiang Yang. Active transfer learning for cross-system recommendation. In *AAAI*, 2013.

[40] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *NIPS*, pages 2110–2118, 2016.

[41] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *ICLR*, 2017.

[42] Peng Xu, Qiyue Yin, Yongye Huang, Yi-Zhe Song, Zhanyu Ma, Liang Wang, Tao Xiang, W Bastiaan Kleijn, and Jun Guo. Cross-modal subspace learning for fine-grained sketch-based image retrieval. *Neurocomputing*, 278:75–86, 2018.

[43] Cheng Wang, Haojin Yang, and Christoph Meinel. Image captioning with deep bidirectional lstms and multi-task learning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 14:40:1–40:20, 2018.

[44] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. Cross domain recommender systems: A systematic literature review. *ACM Comput. Surv.*, 50(3):36:1–36:34, 2017.

[45] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[46] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52, 2015.

[47] Yehuda Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *TKDD*, 4(1):1, 2010.

[48] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *ICLR*, 2014.

[49] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[50] Shulong Tan, Jiajun Bu, Xuzhen Qin, Chun Chen, and Deng Cai. Cross domain recommendation based on multi-type media fusion. *Neurocomputing*, 127:124–134, 2014.

[51] Shuhui Jiang, Zhengming Ding, and Yun Fu. Deep low-rank sparse collective factorization for cross-domain recommendation. In *ACM Multimedia*, pages 163–171, 2017.

[52] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Cheng Wang is a research scientist at NEC Labs Europe. He received his Dr. rer. nat. degree from Hasso Plattner Institute, University of Potsdam (2017). His research interests are machine (deep) learning, multimodal learning with applications in language and vision, information retrieval tasks. He is a PC member of AAAI, IJCAI, NIPS, NAACL, ACL, ACMMM and an invited reviewer of AIJ, IEEE TNNLS, IEEE TIP, IEEE TMM etc.. He is IEEE, ACM and ACL member.

Mathias Niepert is a chief research scientist at NEC Labs Europe. He received his PhD from Indiana University (2009) and was a postdoctoral researcher at the University of Washington, Seattle. His research interests include representation learning for graph-structured data, unsupervised and semi-supervised learning, probabilistic graphical models, and statistical relational learning. He has won several best paper awards as well as research grants such as a Google Research Award. He is a PC member of ICML, NIPS, UAI, ICLR, AAAI and IJCAI.

Hui Li is currently an assistant professor in the School of Information Science and Engineering, Xiamen University. His research interests include data mining and data management with applications in recommender systems and knowledge graph. He received his B.Eng. degree in software engineering from Central South University (2012), and his MPhil and PhD degrees in computer science from the University of Hong Kong (2015, 2018).