DARec: Deep Domain Adaptation for Cross-Domain Recommendation via Transferring Rating Patterns

Feng Yuan , **Lina Yao** and **Boualem Benatallah**

University of New South Wales

feng.yuan@student.unsw.edu.au, lina.yao@unsw.edu.au, b.benatallah@unsw.edu.au

Abstract

Cross-domain recommendation has long been one of the major topics in recommender systems.Recently, various deep models have been proposed to transfer the learned knowledge across domains, but most of them focus on extracting abstract transferable features from auxiliary contents, e.g., images and review texts, and the patterns in the rating matrix itself is rarely touched. In this work, inspired by the concept of domain adaptation, we proposed a deep domain adaptation model (DARec) that is capable of extracting and transferring patterns from rating matrices *only* without relying on any auxillary information. We empirically demonstrate on public datasets that our method achieves the best performance among several state-of-the-art alternative cross-domain recommendation models.

1 Introduction

Recommender systems have long been beset by a host of well-known issues, e.g., the cold-start and sparsity problems. As one solution, cross-domain recommendation (CDR) leverages the information from several different source domains to build a better recommendation model for the target domain. In real-world datasets, although the number of users and items are normally very large with limited available feedbacks, the users/items can be normally categorized into domains according to certain criteria, making CDR a promising technique. CDR has been studied from many perspectives in different research areas [Khan *et al.*, 2017]. Recommendation scenarios can be divided into four classes: 1. No User–No Item overlap (NU-NI); 2. User–No Item overlap (U-NI); 3. No User–Item overlap (NU-I); 4. User–Item overlap (U-I). In this work, we focus on single-domain recommendation task under the U-NI scenario. In other words, we model rating patterns in both source and target domains to enhance the recommendations in the target domain.

Under our research settings, multiple approaches have been proposed. Some are based on clustering [Li *et al.*, 2009], and others use variations of matrix factorization (MF) [Singh and Gordon, 2008]. Recent works apply deep learning (DL) techniques to perform the knowledge transfer [Elkahky *et al.*,

2015; Lian *et al.*, 2017]. Most clustering and MF-based methods fail to model the nonlinear patterns in the ratings and normally require a dense rating matrix for the source domain. DL methods show superior performance to the above ones, but most of them learn and transfer knowledge from auxiliary contents. Directly extracting and transferring patterns from the sparse rating matrix are rarely studied. As DL has been reported to have state-of-the-art performance in rating prediction and top-N recommendation tasks by only using rating matrices [He *et al.*, 2017; He *et al.*, 2018], we propose to leverage deep neural networks (DNN) to learn the transferable rating patterns.

Domain adaptation (DA) is a technique that allows knowledge from a source domain to be transferred to a different but related target domain. DA is widely employed in semi-supervised learning, where the target domain are not labelled or have only a few labels. DA aims to learn a classifier or predictor in presence of a distribution shift between the source and target domain, and has been shown to have state-of-the-art performance in various computer vision and natural language processing tasks [Bousmalis *et al.*, 2016].

Typical recommendation problems do not exactly match the scenarios of DA. However, as a fundamental question in cross-domain recommendations, how to model the data-dependent effect of the ratings when transferring the data-independent knowledge needs to be addressed. As the source and target data may be with different distributions, inspired by the domain adversarial neural networks (DANN) [Ganin and Lempitsky, 2015], we propose a deep Domain Adaptation Recommendation (DARec) model. It contains a rating pattern extractor, a domain classifier and a rating predictor. It can learn abstract representations of shared patterns and transfer them between two domains via DNN. Furthermore, we apply a weighted loss function to balance the significance of the source and target domains to control the transfer direction. Due to the sparsity in the rating matrices, we first apply AutoRec [Sedhain *et al.*, 2015] to learn a dense embedding for each user to represent the user's preferences, after which the embeddings are fed into the following DNN. With DARec, the rating estimation accuracy in the target domain can be improved. We further compare DARec with several state-of-the-art CDR methods on public datasets, showing that DARec achieves best recommendation results. Our major contributions are as follows:

- We propose a deep domain adaptation cross-domain recommendation model (DARec) to extract and transfer abstract rating patterns, using only the information from rating matrices. Utilizing a domian classifier, shared rating patterns of the same user in different domains are learned and transferred via adversarial training, leading to U-DARec model, while distinctive rating patterns of unrelated items in different domains are separated, resulting in the I-DARec model;

- Using serveral public datasets, we empirically investigate the capability of our approach to improve on the recommendation performance in single-domain, which confirms the effectiveness of DARec to transfer knowledge between domains. We compare our model with some state-of-the-art single-domain rating prediction methods and demonstrate that benefiting from the transferred information from the source domain, DARec is superior to all the selected baselines;

- We select several alternative cross-domain recommendation methods that can fit in our research settings, including MF-based models and recently-proposed DL-based ones. On the public datasets, we demonstrate that due to DL, DARec is superior to all the selected MF models, and I-DARec exceeds the performance of state-of-the-art DL-based methods.

2 Related Work

Generally, CDR can be categorized as collaborative filtering-based CDR (CFCDR) methods and content-based CDR (CBCDR) methods.Early CFCDR approaches use neighborhood-based solutions [Berkovsky *et al.*, 2007], but these methods are soon surpassed by MF-based solutions. Collective Matrix Factorization (CMF) [Singh and Gordon, 2008] couples target rating matrix and all auxiliary matrices on *User* dimension to share the user factor matrix across all domains. Cross-domain Triadic Factorization (CDTF) [Hu *et al.*, 2013] learns the triadic factors for user, item and domain through tensor factorization, which can better capture the interactions between domain-specific user factors and item factors. Cluster-level Matrix Factorization [Mirbakhsh and Ling, 2015] employs SVD and K-means to build a cross-domain cluster-level coarse matrix that captures shared patterns between the cluster of users and the cluster of items among domains. Most of these models have three major shortcomes: 1. most of them assume the auxiliary data is relatively dense for all users or items; 2. user/item overlaps are normally required; 3. the linear MF methods are not powerful enough to extract nonlinear patterns .

To overcome such limitations, CBCDR techniques establish domain links by leveraging the auxiliary information about users/items without requiring user/item overlaps or relatively dense rating matrices. Moreover, DL is ready to be applied on auxiliary data to better represent users/items. [Elkahky *et al.*, 2015] is a multi-view framework that extends the deep structured semantic model which maps high-dimensional features into low-dimensional dense features in a joint semantic space. [Lian *et al.*, 2017] formulates a multi-view neural network learning framework that combines MF-extracted features with user/item features. [Kanagawa *et al.*, 2018] proposes a DA-based model, where stacked denoising autoencoders are used to extract item features, and a domain separation network is responsible for recommendation. The above methods all rely on auxiliary contents to learn shared patterns while emphasize less on the patterns in the rating matrices. Recently, a work named collaborative cross networks (CoNet) [Hu, 2018] employs deep cross connection neural networks to learn and transfer shared rating patterns among domains without using any auxiliary contents.

Unlike all the previous works, our approach is different in four perspectives:1. we leverage DNN to learn linear and nonlinear shared rating patterns from rating matrices, which is more powerful than MF- or clustering-based methods; 2. no dense data from the source domain is required for extracting transferable knowledge, as we rely on DNN to learn such knowledge from the raw sparse rating matrices; 3. we do not need any auxiliary user/item features; 4. we combine DA in our approach so that the possible distribution mismatch between two domains can be alleviated, which is more effective than simply using fully-connected neural networks to model the difference between two domains as CoNet .

3 Proposed Method

3.1 Problem Definition

In this paper, we assume that the input data takes the form of explicit feedbacks such as user ratings of items. We also assume that the source and target domains have the same set of users (i.e. the U-NI scenario), denoted by $\mathcal{U} = \{1, 2, ..., U\}$. The item sets from the source and target domains are $\mathcal{I}^{\mathcal{S}} = \{1, 2, ..., I^S\}$ and $\mathcal{I}^{\mathcal{T}} = \{1, 2, ..., I^T\}$ respectively, with the corresponding rating matrices given by $\mathcal{Y}^{\mathcal{S}} = \{y_{ui}^S | u \in \mathcal{U}, i \in \mathcal{I}^{\mathcal{S}}\}$ and $\mathcal{Y}^{\mathcal{T}} = \{y_{ui}^T | u \in \mathcal{U}, i \in \mathcal{I}^{\mathcal{T}}\}$. We denote the set of observed item ratings given by u as $\mathcal{I}_u^S$ ($\mathcal{I}_u^T$), and the unobserved ones as $\bar{\mathcal{I}}_u^S$ ($\bar{\mathcal{I}}_u^T$) for source(target) domain. In each domain, the goal of recommendation can be specified as selecting a subset of items from $\bar{\mathcal{I}}_u^S$ ($\bar{\mathcal{I}}_u^T$) for user u according to certain criteria that maximizes the user's satisfaction. In other words, the recommendation algorithms aim to give predictions on the unknown ratings of each user, i.e., $\hat{y}_{ui}^S (u \in \mathcal{U}, i \in \bar{\mathcal{I}}_u^S)$ or $\hat{y}_{ui}^T (u \in \mathcal{U}, i \in \bar{\mathcal{I}}_u^T)$. The values of y_{ui}^S (y_{ui}^T) are numerical ratings in a certain range, e.g. $[1, 5]$. The recommendation task we consider here is the single-domain problem [Khan *et al.*, 2017], where we predict $\hat{y}_{ui}^T (u \in \mathcal{U}, i \in \bar{\mathcal{I}}_u^T)$, via leveraging the information from the source rating matrix $\mathcal{Y}^{\mathcal{S}}$. We use the widely-adopted metrics to measure the performance of all approaches used in this paper, such as RMSE defined by $\sqrt{1/(MN) \sum_{u=1}^M \sum_{i=1}^N (\hat{y}_{ui}^T - y_{ui}^T)^2}$, where M,N denote the number of users and items in the testing set respectively.

3.2 Deep Domain Adaptation for CDR

Figure 1 shows the proposed model to integrate DA for CDR. When the source and target domains share the same set of users, but different items, we use AutoRec to learn a set of embeddings to represent each user's preferences. AutoRec
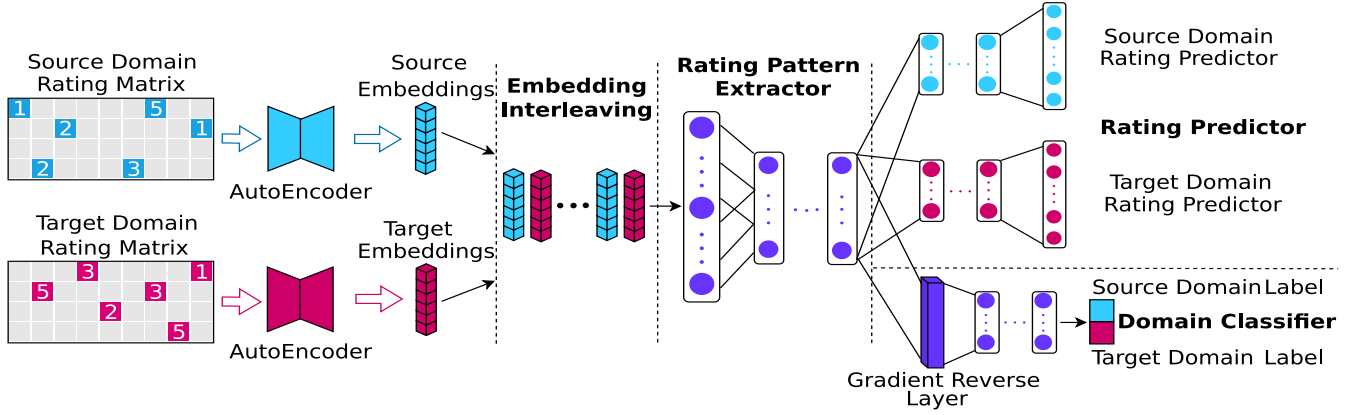
Figure 1. The Structure of Deep Domain Adaptation-Based Cross-Domain Recommendation

[Sedhain *et al.*, 2015] leverages an autoencoder (AE) to predict the missing values in the rating matrix. U-AutoRec (I-AutoRec) first takes the partially observed rating vectors for each user (item), i.e. $\mathbf{y_u}$ ($\mathbf{y_i}$) as input, and maps each vector into a low-dimensional latent space, followed by a reconstruction layer as output to recover the rating vectors so that the original missing ratings can be generated for recommendation purpose. Formally, the reconstructed vectors are written as:

$$\hat{\mathbf{y}} = h(\mathbf{W_2}\, g(\mathbf{W_1}\, \mathbf{y} + \mathbf{b_1}) + \mathbf{b_2}) \qquad (1)$$

where $h(\cdot)$ and $g(\cdot)$ are activation functions, $\mathbf{W_1}, \mathbf{W_2}$, $\mathbf{b_1}$, $\mathbf{b_2}$ are corresponding weights and biases of the AE. The loss function is regularized square loss (U-AutoRec):

$$loss_{AutoRec} = \sum_{u=1}^{U} \|\hat{\mathbf{y}}_{\mathbf{u}} - \mathbf{y_u}\|_{\mathcal{O}}^2$$
$$+\alpha(\|\mathbf{W_1}\|_F^2 + \|\mathbf{W_2}\|_F^2 + \|\mathbf{b_1}\|_2^2 + \|\mathbf{b_2}\|_2^2) \qquad (2)$$

where $\|\cdot\|_F$ and $\|\cdot\|_2$ denote matrix Frobenius norm and vector l^2-norm respectively, $\mathcal{O}$ denotes the observed ratings in each vector $\mathbf{y_u}$, α controls the regularization strength.

After training the AutoRec, we calculate $\boldsymbol{f_u} = g(\mathbf{W_1}\, \mathbf{y_u} + \mathbf{b_1})$ with the trained parameters $\mathbf{W_1}$ and $\mathbf{b_1}$, as the low-dimensional latent factors for each user in both domains. $\boldsymbol{f_u}$ are then fed into a modified DANN network. To better extract the shared rating patterns from the latent space, the sequence of the latent factors for target domain are interleaved by those from the source domain. To modify the original DANN [Ganin and Lempitsky, 2015] for rating prediction recommendation task, we use one deep feed-forward neural network with parameter set Θ_r to reconstruct the rating vectors for source and target domain separately (rating predictor) and another neural network with parameter set Θ_c to predict the corresponding domain label $c \in \{0, 1\}$, where 0 denotes the input vector belongs to the source domain and 1 the target domain (domain classifier). A fully-connected neural network is applied for feature extraction with parameter set Θ_f (rating pattern extractor). The training procedure to find the domain-invariant features involves optimizing on Θ_f to maximize the loss of the domain classifier, while optimizing on

Θ_c and Θ_r to minimize the loss of the domain classifier and the label predictor. The total loss function is the combination of the predictor loss and the classifier loss:

$$L(\Theta_f, \Theta_r, \Theta_c) = loss_{pre}(\Theta_f, \Theta_r) - \mu\, loss_{dom}(\Theta_c) + \lambda R$$
$$= \sum_{u=1}^{2U} \|\hat{\mathbf{y}}_{\mathbf{u,S}} - \mathbf{y_{u,S}}\|_{\mathcal{O}}^2 + \beta\|\hat{\mathbf{y}}_{\mathbf{u,T}} - \mathbf{y_{u,T}}\|_{\mathcal{O}}^2$$
$$+ \mu \sum_{u=1}^{2U} \hat{c}_u \log(c_u) + (1 - \hat{c}_u)\log(1 - c_u)$$
$$+ \lambda R$$
$$(3)$$

where we employ binary cross-entropy for domain label classification loss ($\hat{c}_u$ is the predicted domain and c_u is the ground-truth). $R = \|\Theta_f\|^2 + \|\Theta_r\|^2 + \|\Theta_c\|^2$ is the regularizer term to alleviate overfitting, in which λ controls the regularization strength. β is used to balance the significance of the source and target parts in the total loss function. μ controls the portion of loss contributed by the domain classifier. Then the optimization objective can be expressed as:

$$(\hat{\Theta}_f, \hat{\Theta}_r) = \underset{\Theta_f, \Theta_r}{\arg\min}\; L(\Theta_f, \Theta_r, \Theta_c)$$
$$\hat{\Theta}_c = \underset{\Theta_c}{\arg\max}\; L(\Theta_f, \Theta_r, \Theta_c) \qquad (4)$$

To solve the above problem by stochastic grandient desent (SGD)-like algorithms, the original DANN introduces the gradient reversal layer (GRL). GRL is a simple structure that has different behaviours for forward and back propagation with no intrinsic parameters. In forward propagation, GRL acts as an identity function to let data pass through, while in backpropagation, it calculates the gradient from the subsequent level, multiplies it by $-\mu$ and then passes it to the preceding layer. Formally, GRL can be treated as a "function":

$$\Phi(\boldsymbol{x}) = \boldsymbol{x} \qquad (forward\ propagation)$$
$$\frac{d\Phi(\boldsymbol{x})}{d\boldsymbol{x}} = -\mu\mathbf{I} \qquad (back\ propagation) \qquad (5)$$

where $\mathbf{I}$ is the identity matrix. GRL is placed immediately after the rating pattern extractor and before the domain clas-

sifier. Therefore, the total loss function adapted to SGD-like algorithm is converted as:

$$\tilde{L}(\Phi, \Theta_f, \Theta_r, \Theta_c) = loss_{pred}(\Theta_f, \Theta_r) \\ + loss_{dom}(\Phi, \Theta_c) + \lambda R \quad (6)$$

Equation 4 is designed for extracting the deep shared rating patterns for each user in the target and source domain, which is called U-DARec. Our approach can also be applied on obtaining patterns from each item rating vector. We first use I-AutoRec to train a set of latent factors for all items in both domains, and feed them into the following DANN. As there is no overlap for items in the two domains, here, we extract the domain-exclusive rating patterns from the raw rating vectors and separate them using the rating predictor. Thus, the loss of the domain classifier is minimized, leading to a different optimization objective from Equation 4:

$$(\hat{\Theta}_f, \hat{\Theta}_r, \hat{\Theta}_c) = \underset{\Theta_f, \Theta_r, \Theta_c}{\arg\min} \ L(\Theta_f, \Theta_r, \Theta_c) \quad (7)$$

Consequently, the GRL layer is no longer needed and SGD-like algorithms can directly optimize Equation 7. We call this model I-DARec.

To train DARec, we apply mini-batch Adam algorithm for both user embedding training and rating prediction procedures. In AutoRec, we initialize W_1, W_2 with normal distribution (zero mean, 0.01 standard deviation), b_1, b_2 with zeros and randomly draw a mini-batch $\mathcal{U}'$ with size B from $\mathcal{U}$ to obtain the embeddings. Then, the calculated f_u (f_i for I-DARec) in both domains are interleaved with each other, and are subsequently fed into the modified DANN. Similarly, we initialize the weights in DANN with normal distribution (zero mean, 0.01 standard deviation), and biases with zeros. Then, we train the parameters $\hat{\Theta}_f, \hat{\Theta}_r, \hat{\Theta}_c$ in Equation 4 or Equation 7 for U-DARec and I-DARec respectively.

4 Experiments

In this section, we systematically evaluate the DARec model on multiple subsets extracted from the Amazon dataset with shared users in different categories. After giving the detailed dataset settings, we investigate the effectiveness of knowledge transfer in our approach and compare it with several state-of-the-art methods for cross-domain recommendation.

4.1 Experimental Settings

Datasets
We use the public dataset collected by J. McAuley [He and McAuley, 2016] and define different item categories as domains, where we select users with at least 5 ratings. We deliberately select categories that are as irrelevant as possible and the statistics are summarized in table 1. Both the source and target domains are extremely sparse with at least 99.8% of the ratings are unobserved, which poses a major challenge on most clustering-based and MF-based CDR methods that normally require a relatively dense source domain.

Parameter Settings
In the embedding training stage, we leave out 10% of the data as validation set to tune the hyperparameters where we adjust the number of hidden neurons from 100 to 1500 and regularizer coefficient from 0.1 to 0.00001 to get the lowest RMSE that can be achieved by AutoRec. Then, we feed all the data into the pre-trained AutoRec to get the embeddings calculated as f_u (f_i). After the sequence of embeddings from source domain is interleaved with those from the target domain, we send it into the DANN. For the rating pattern extractor, we use only one layer with hidden neurons varying from 50 to 500. In the rating predcitor, we apply 3 layers for each domain and 2 layers for the domain classifier, where the number of neurons in each layer follows a pyramid shape and varies with the dimension of the previous pattern extractor. The parameter β is changed from 0.0001 to 1 so that the source domain is emphasized, and μ, λ are varied from 0.0001 to 10,000. To process the dataset, we randomly leave out $10\%(20\%)$ of the data for testing and $90\%(80\%)$ for training. 10% of the training set is used as validation set for hyperparamater tuning.

4.2 Effectiveness of Transfer Learning
We first investigate whether DARec can enhance the recommendation accuracy in the target domain. Apart from the standard AutoRec, we compare our approach with several state-of-the-art single-domain rating prediction models:

- **PMF** [Mnih and Salakhutdinov, 2008] Probabilistic Matrix Factorization is one of the most populare MF methods. It obtains the predicted ratings from the inner product of corresponding user and target item latent factors.

- **RBM** [Salakhutdinov et al., 2007] Restricted Boltzmann Machine uses neural networks to reconstruct the unknown ratings from the observed ones.

- **AutoRec** [Sedhain et al., 2015] AutoRec is an RBM-like model which replaces the RBM with an AE to perform the unknown rating reconstruction procedure.

- **CF-NADE** [Zheng et al., 2016] Neural Autoregressive Distribution Estimator is applied for CF tasks where it replaces the role of RBM for rating reconstruction.

For PMF, we use the package Suprise[1]. For RBM, we rewrote a tensorflow version according to the opensource implementation[2] and for CF-NADE, we choose the version of I-CF-NADE-S and rewrote a tensorflow version with reference to the opensource implementation[3]. We tune the latent factor dimensions, learning rate and regularizer coefficients to get their best performance.

Figure 3 shows the average RMSE values of the compared single-domain recommendation methods, from which we can observe that both U-DARec and I-DARec obtain obvious improvment over U-AutoRec and I-AutoRec respectively. In U-DARec, the shared rating patterns are transferred via the DNN, while in I-DARec, the distinct rating patterns in each domain are extracted. For instance, in the Office Products and Movies & TV dataset (No.1), U-DARec improves 3.66% in RMSE compared with U-AutoRec and in the Sports & Outdoors and CDs & Vinyl dataset (No.2), the improvement is

---

[1]https://github.com/NicolasHug/Surprise
[2]https://github.com/felipecruz/CFRBM
[3]https://github.com/Ian09/CF-NADE

| No.	Dataset		Item#		User#		Ratings#		Sparsity	
	Source	Target	Source	Target	Shared	Source	Target	Source	Target
1	Office Products	Movies and TV	10,398	21,732	5,154	40,294	158,927	99.92%	99.86%
2	Sports and Outdoors	CDs and Vinyl	16,420	34,286	5,713	39,151	79,019	99.99%	99.96%
3	Android Apps	Video Games	9,185	10,062	2,034	34,217	21,312	99.82%	99.90%
4	Toys and Games	Automotive	10,597	7,375	2,885	25,103	16,448	99.92%	99.92%

Table 1. Statistics of the Datasets
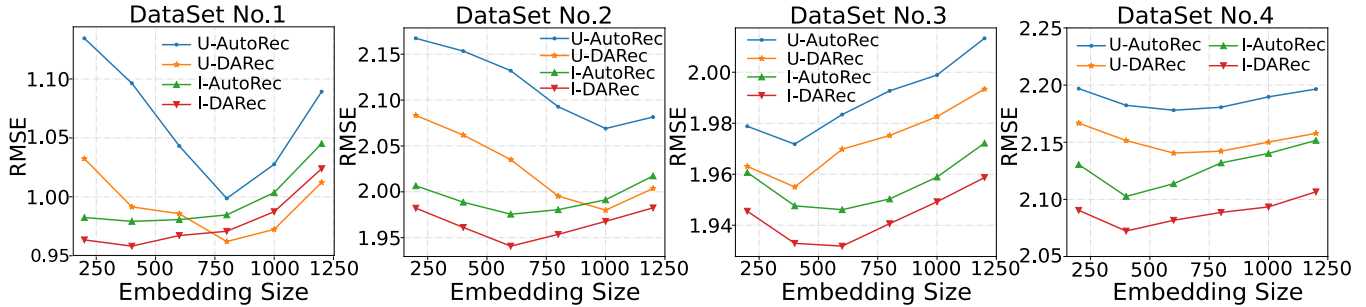
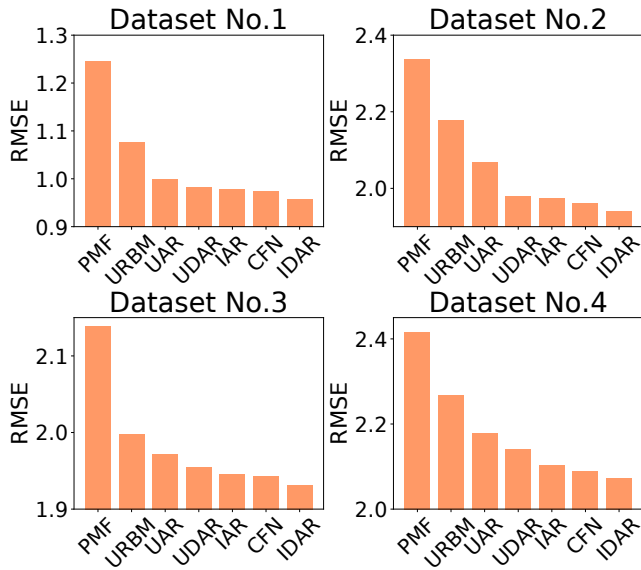Figure 2. Impact of Embeddng Size on Prediction Accuracy

Figure 3. Comparison with Single-domain Approaches (URBM: U-RBM, UAR: U-AutoRec, UDAR: U-DARec, IAR: I-AutoRec, CFN: CF-NADE, IDAR: I-DARec)

4.30%. Although U-DARec still can not compete with I-AutoRec and CF-NADE, yet we achieve best performance by further improving on I-AutoRec via I-DARec. The superior performance from I-DARec is probably due to two reasons: 1. the larger number of different input embeddings for I-DARec than that for U-DARec, making the training more effective for I-DARec; 2. the adversarial objectives in U-DARec pose a trade-off between extracting shared patterns and enhancing rating prediction accuracy.

We then investigate the impact of the embedding size

trained by AutoRec on the single-domain recommendation performance. We first fix the structure of the DANN(i.e., the number of hidden neurons of rating pattern extractor is set to 100) and then change the dimension of the embeddings from 200 to 1200 for all datasets. From figure 2, it can be observed that for both U-DARec and I-DARec, there exist optimal embedding sizes. For instance, in the Office Products and Movies & TV dataset (No.1), U-DARec reaches minimum RMSE at around 800 and I-DARec at around 400. When the input embedding dimension is not large enough, the rating patterns can not be effectively extracted, while at a large embedding size, the learned embeddings are too noisy with undesired latent feature details, bringing detrimental effects on the overall performance. Furthermore, we also notice that although I-DARec achieves the lowest RMSE in general, yet the enhancement from U-DARec is greater than that from I-DARec, which can be well understood from the aligned-user settings of the datasets as the model is more effective in learning the shared rating patterns.

4.3 Comparison with Baselines

- **CMF** [Singh and Gordon, 2008] Collective Matrix Factorization (CMF) is a latent factor model that establishes relationships among multiple domains via MF. Our experiment settings correspond to the three-entity-type schema with two domains. The first entity type is the items in the source domain and the third entity type is the items in the target domain, with the shared users as the bridging entity type. A set of latent factors is learned for each entity type and rating predictions are made via inner products of corresponding latent factors. It is a MF-based transfer learning approach that jointly factorizes two related domains.

- **CDTF** [Hu *et al.*, 2013] Cross Domain Triadic Factor-

Dataset	Office Products & Movies and TV		Sports and Outdoors & CDs and Vinyl		Android Apps & Video Games		Toys and Games & Automotive	
%	80	90	80	90	80	90	80	90
CMF	1.0825	1.0583	2.1043	2.0578	2.0784	2.1295	2.2636	2.1855
CDTF	1.0577	1.0389	2.0702	2.0126	2.0461	1.9965	2.2234	2.1680
FM-CDCF	1.0272	0.9925	2.0580	1.9942	1.9963	1.9722	2.1908	2.1534
CoNet	0.9782	0.9689	1.9835	1.9587	1.9744	1.9562	2.1507	2.1356
U-DARec	0.9985	0.9821	2.0123	1.9808	1.9876	1.9550	2.1734	2.1405
I-DARec	**0.9665**	**0.9582**	**1.9673**	**1.9408**	**1.9546**	**1.9318**	**2.1152**	**2.0723**

Table 2. Comparison with Baselines (%: percentage of training data)

ization (CDTF) is another latent factor model that leverages the triadic relation user-item-domain. It learns the triadic factors for user, item and domain via factorizing a three-order tensor with weighted square loss, where each slice of the tensor corresponds to one domain. CDTF matches the CDR scenario we consider, where each domain has equivalent number of users but vary in the number of items.

- **FM-CDCF** [Loni *et al.*, 2014] Factorization Machine Cross-Domain Collaborative Filtering (FM-CDCF) is based on FM, where the feature vector in single-domain FM is extended to incorporate collaborative information from other domains. It is well suited for the aligned-user CDR scenario where each input feature vector is composed of the one-hot representation of the user u, one-hot representation of the item i in the target domain, and all the normalized ratings given by u in the source domain. It is still a shallow model without the ability of capturing nonlinear interactions between domains.

- **CoNet** [Hu, 2018] Collaborative Cross Networks (CoNet) is a state-of-the-art deep CDR model that transfers knowledge between domains with aligned users, which matches our research settings. It uses DNN to learn latent features in each domain and applies a modified cross-stitch neural network to enable knowledge transfer. It is a highly competitive method and achieves best performance among other non-DL approaches.

For CMF, we rewrote a python version based on the original Matlab code[4]. For FM-CDCF, we adopt the libfm implementation[5] for FM and feed in the extended input feature vectors. For CDTF and CoNet, we implement our own python version according to the original papers. For all the baselines, we tune the latent factor dimensions, learning rate and regularizer coefficients to get their best performance.

Table 2 reports the comparison results among the selected baselines for CDR. Note that in each domain pair, the chosen categories are as irrelevant as possible, so that we can better examine the performance of our model on learning and transfering knowledge from the rating matrix only. It is ob-

vious that DL-based approaches (i.e., CoNet and DARec) are superior to non-deep models. This can be explained by the ability of nonlinearity modelling from the deep neural networks. Also, MF-based mehtods perform better when the source domain dataset is relatively dense, but the datasets we consider in this work are of high sparsity (over 99.8%). For example, in the Office Products and Movies & TV dataset with 90% training data, CoNet gains 2.38% RMSE improvement over FM-CDCF and I-DARec achieves 3.45% over FM-CDCF. This result shows that DL-based methods are more suitable for extreme sparse datasets, which is in consistency with the findings in [Hu, 2018]. Furthermore, we notice that CoNet performs better than U-DARec as U-DARec adopts adversarial training to balance the extraction of shared user rating patterns and the pursuit of high rating prediction accuracy. This weakness is overcome by I-DARec which minimizes the losses of domain classifier and rating predictor at the same time. Thus, as we can see, I-DARec achieves the best performance among the baselines.

5 Conclusions

In this paper, we proposed the DARec model for cross-domain recommendation, where we consider the single-domain recommendation task with users aligned and only rely on the rating matrices. To extract the shared user rating patterns from the two domains, we first apply AutoRec to generate embeddings to represent each user. Then, the embeddings from the source and target domains are interleaved and fed into DANN for knowledge transfer. Using a domain classifier, the shared deep rating patterns from the two domains are extracted, followed by a rating predictor that separates the estimated ratings for each domain. We further extend our model by taking the ratings for each item as input to extract the distinct features of the two items. We demonstrate that our model achieves best performance in rating prediction task among several widely-adopted and state-of-the-art cross-domain recommendation approaches. In the future, we will integrate content-based domain adaptation models to gain further performance improvement.

---

[4]http://www.cs.cmu.edu/ ajit/cmf/

[5]http://www.libfm.org/

References

[Berkovsky *et al.*, 2007] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Cross-domain mediation in collaborative filtering. In *International Conference on User Modeling*, pages 355–359. Springer, 2007.

[Bousmalis *et al.*, 2016] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016.

[Elkahky *et al.*, 2015] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288. International World Wide Web Conferences Steering Committee, 2015.

[Ganin and Lempitsky, 2015] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[He and McAuley, 2016] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[He *et al.*, 2018] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912*, 2018.

[Hu *et al.*, 2013] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 595–606. ACM, 2013.

[Hu, 2018] Herbert Hu. Conet: Collaborative cross networks for cross-domain recommendation. *arXiv preprint arXiv:1804.06769*, 2018.

[Kanagawa *et al.*, 2018] Heishiro Kanagawa, Hayato Kobayashi, Nobuyuki Shimizu, Yukihiro Tagami, and Taiji Suzuki. Cross-domain recommendation via deep domain adaptation. *arXiv preprint arXiv:1803.03018*, 2018.

[Khan *et al.*, 2017] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. Cross domain recommender systems: a systematic literature review. *ACM Computing Surveys (CSUR)*, 50(3):36, 2017.

[Li *et al.*, 2009] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th annual international conference on machine learning*, pages 617–624. ACM, 2009.

[Lian *et al.*, 2017] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. Cccfnet: a content-boosted collaborative filtering neural network for cross domain recommender systems. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 817–818. International World Wide Web Conferences Steering Committee, 2017.

[Loni *et al.*, 2014] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. Cross-domain collaborative filtering with factorization machines. In *ECIR*, 2014.

[Mirbakhsh and Ling, 2015] Nima Mirbakhsh and Charles X Ling. Improving top-n recommendation for cold-start users via cross-domain information. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):33, 2015.

[Mnih and Salakhutdinov, 2008] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[Salakhutdinov *et al.*, 2007] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[Sedhain *et al.*, 2015] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, pages 111–112. ACM, 2015.

[Singh and Gordon, 2008] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

[Zheng *et al.*, 2016] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. A neural autoregressive approach to collaborative filtering. *arXiv preprint arXiv:1605.09477*, 2016.