Domain-Invariant Task Optimization for Cross-domain Recommendation

Dou Liu[1,2,3], Qingbo Hao[1,2,3], Yingyuan Xiao[1,2,3]✉, Wenguang Zheng[1,2,3] and Jinsong Wang[1,2,3]

[1] School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China
[2] Engineering Research Center of Learning-Based Intelligent System, Ministry of Education, Tianjin University of Technology, Tianjin 300384, China
[3] Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin 300384, China
yyxiao@tjut.edu.cn

Abstract. The challenge of cold start has long been a persistent issue in recommender systems. However, Cross-domain Recommendation (CDR) provides a promising solution by utilizing the abundant information available in the auxiliary source domain to facilitate cold-start recommendations for the target domain. Many existing popular CDR methods only use overlapping user data but ignore non-overlapping user data when training the model to establish a mapping function, which reduces the model's generalization ability. Furthermore, these CDR methods often directly learn the target embedding during training, because the target embedding itself may be unreasonable, resulting in an unreasonable transformed embedding, exacerbating the difficulty of model generalization. To address these issues, we propose a novel framework named Domain-Invariant Task Optimization for Cross-domain Recommendation (DITOCDR). To effectively utilize non-overlapping user information, we employ source and target domain autoencoders to learn overlapping and non-overlapping user embeddings and extract domain-invariant factors. Additionally, we use a task-optimized strategy for target embedding learning to optimize the embedding and implicitly transform the source domain user embedding to the target feature space. We evaluate our proposed DITOCDR on three real-world datasets collected by Amazon, and the experimental results demonstrate its excellent performance and effectiveness.

Keywords: Cross-domain Recommendation, Cold-start Recommendations, Task Optimization.

1 INTRODUCTION

The development of the network world creates a challenge in extracting useful information from vast data, making recommender systems crucial for information filtering. However, collaborative filtering and deep learning models often struggle with the cold start problem in real-world scenarios due to data sparsity. Cross-domain recommendation (CDR) [1-5] addresses this issue by leveraging user interactions across multiple fields, such as ratings, articles, and videos. CDR aims to transfer information from the auxiliary domain to the target domain, alleviating the cold start problem.

Most existing CDR approaches employ embedding and mapping methods. Initially, user preferences in the source and target domains are encoded separately as embeddings. An explicit mapping function is then learned based on overlapping users to minimize the distance between the target embedding and the mapped embeddings.

Previous research indicates a significant challenge faced by most embedding and mapping models: the low proportion of overlapping users between the source and target domains, typically less than 10% in real-world scenarios. Our experiments involve two tasks, with overlap rates of 9.95% for Task 1 and only 5.41% for Task 2, as shown in Table 1. This limitation hinders model generalization and impairs the performance of cold-start users in the target domain. Additionally, the mapping method learns to map user embeddings from the source domain to the target domain, aiming to minimize the distance. However, the target domain user embeddings are obtained through encoding learning, potentially leading to unreasonable target embeddings and further complicating the challenge of model generalization.

Table 1. Statistics of the tasks datasets (#*Rate* denotes the overlap ratio)

	#Domain	#Users	#Items	#Ratings	#Overlap	#Rate
Task1	Movie	123,960	50,052	1,697,533	18,031	9.95%
	Music	75,258	64,443	1,097,592		
Task2	Book	603,668	367,982	8,898,041	37,388	5.41%
	Movie	123,960	50,052	1,697,533		

To address these challenges, we propose a novel method called Domain-Invariant Task Optimization for Cross-domain Recommendation (DITOCDR). DITOCDR effectively tackles the cold-start problem caused by data sparsity by utilizing non-overlapping user data and training task-oriented methods to optimize the learning process of target embeddings, resulting in improved model performance. Experimental results demonstrate the practicality and generalization ability of DITOCDR.

The main contributions of our work can be summarized in three aspects:

• We introduce DITOCDR, a domain-invariant task optimization framework, to tackle the cold-start problem in CDR. DITOCDR employs autoencoder networks in both the source and target domains to learn embedding information for all users, overcoming the generalization challenge caused by a low number of overlapping users.

• We utilize a task-oriented optimization approach that focuses on rating rather than mapping. This technique mitigates the negative impact of unreasonable user embeddings, leading to improved recommendation accuracy.

• We adopt a joint training approach that simultaneously trains multiple tasks mentioned above. This approach involves shared user embedding parameters, mutually regularized tasks, and implicit data augmentation. Experimental results demonstrate the effectiveness of our method in addressing cold-start scenarios.

2 RELATED WORK

2.1 Cross-domain Recommendation

According to CMF[6] assumes that user embeddings in different domains can be shared to establish a global user embedding matrix. In recent years, several models utilizing deep learning methods have been proposed for Cross-domain Recommendation. Prior work relevant to ours is based on cross-domain embedding mapping methods[2-5, 7-11]. EMCDR[3] is a commonly used CDR model that learns a mapping function as a bridge for user embeddings to map from the source domain to the target domain. SSCDR[2] uses a semi-supervised method and encodes distance information according to unlabeled data when learning the mapping function. LACDR[5] optimizes user embedding mapping by utilizing non-overlapping user data. DCDCSR[4] considers rating sparsity of individual users and items in different domains when learning the mapping function. PTUPCDR[11] learns user feature embeddings through a meta-network to generate personalized bridge functions for each user, achieving personalized preference transfer. However, PTUPCDR, like existing embedding mapping-based methods, requires training the mapping function using overlapping users and does not leverage non-overlapping user data.

The main difference between our work and previous literature is that we not only effectively utilized non-overlapping user data to learn cross-domain mapping but also optimized the task of the mapped user embeddings.

3 APPROACH

3.1 Problem Preliminary

In the CDR setting, two distinct domains are present: the source domain, also known as the auxiliary domain, and the target domain. Each domain comprises a set of users $\boldsymbol{\mathcal{U}} = \{u_1, u_2, \dots\}$, a set of items $\boldsymbol{\mathcal{V}} = \{v_1, v_2, \dots\}$, and a rating matrix, denoted by $r_{ij} \in \mathcal{R}$ which represents the rating assigned by a user to an item, signifying the interaction between them, and typically expressed as an integer, such as a scale from 1 to 5. Specifically, $\boldsymbol{\mathcal{U}^s}$, $\boldsymbol{\mathcal{V}^s}$, $\boldsymbol{\mathcal{R}^s}$ represent the user set, item set, and rating matrix of the source domain, respectively, while $\boldsymbol{\mathcal{U}^t}$, $\boldsymbol{\mathcal{V}^t}$, $\boldsymbol{\mathcal{R}^t}$ represent the user set, item set, and rating matrix of the target domain. An overlapping user set between the two domains exists, which is defined as $\boldsymbol{\mathcal{U}^o} = \boldsymbol{\mathcal{U}^s} \cap \boldsymbol{\mathcal{U}^t}$. In this context, a cold start user refers to a user who has not interacted in the target domain, but has interacted in the auxiliary domain. Our main objective is to provide effective recommendations to cold start users in the target domain. The detailed description of the overall algorithm of DITOCDR can be found in Algorithm 1.

Algorithm 1 Domain-Invariant Task Optimization for CDR (DITOCDR)

Input: $\mathcal{U}^s$, $\mathcal{U}^t$, $\mathcal{V}^s$, $\mathcal{V}^t$, $\mathcal{U}^o$, $\mathcal{R}^s$, $\mathcal{R}^t$.
Input: task optimization network g_θ.
Pre-training Stage:
1. A source model contains u^s, v^s.
2. A target model contains u^t, v^t.

Extracting Domain Invariant Factors Stage:
3. Learning source autoencoder and target autoencoder and extracting domain invariant factors by minimizing Equation (6), (7) and (8).

Task-based Optimization Stage:
4. Learning a task optimization network g_θ by minimizing Equation (10).

Test Stage:
5. for a cold-start user u, we use $g_\theta(\mathrm{u}^s)$ as the user embedding for prediction.

We will explain in detail below. The overall structure of DITOCDR is shown in Figure 1.

3.2 Pre-training

Pre-training involves training a latent factor model for both the source and target domains. This model converts users and items into high-dimensional vectors, where each user and item has a distributed representation in this high-dimensional space. These vectors are commonly known as word embeddings or embedding vectors, and are considered as trainable parameters. The two most common methods for training latent factor models are matrix factorization (MF) and deep learning-based methods. MF decomposes the rating matrix into two low-dimensional matrices, while deep learning-based methods use a multi-layer perceptron for training. In this paper, R denotes the rating matrix, d represents the dimensionality of latent factors, $\boldsymbol{u}_i^s \in \mathbb{R}^d$ and $\boldsymbol{v}_j^s \in \mathbb{R}^d$ represent the user embedding and item embedding in the source domain, respectively, and $\boldsymbol{u}_i^t \in \mathbb{R}^d$ and $\boldsymbol{v}_j^t \in \mathbb{R}^d$ represent the corresponding embeddings in the target domain. Through Gaussian observation noise, the probability of the observed rating r_{ij} for a user's evaluation of an item can be modeled.

$$P(r_{ij} | u_i, v_j; \sigma^2) = N(r_{ij} | u_i^T v_j, \sigma^2). \tag{1}$$

Subsequently, the user and item embedding parameters were estimated using maximum likelihood estimation, which requires minimizing the following loss function:

$$(u_i v_j, r_{ij}) = \frac{1}{|\mathcal{R}|} \sum_{r_{ij} \in \mathcal{R}} (r_{ij} - u_i v_j)^2. \tag{2}$$

where r_{ij} denotes the ground truth label and $|\mathcal{R}|$ denotes the number of ratings.

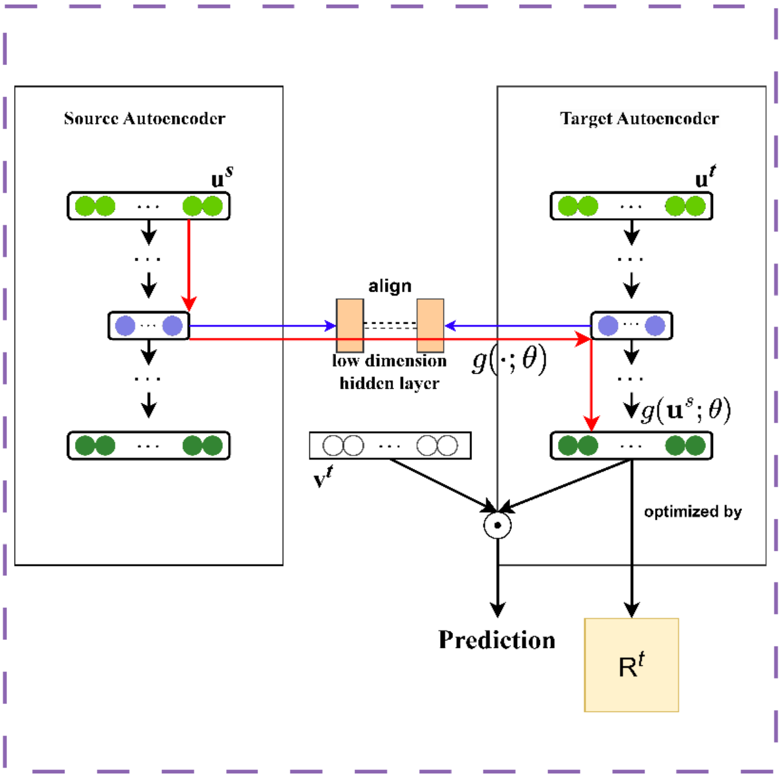


Fig. 1. Illustrative diagram of the DITOCDR framework.

3.3 Autoencoder Network

The embedding-based approach requires the user embeddings of the source domain and a mapping function that maps the user embeddings from the source domain to the target domain. Typically, the mapping function is trained by minimizing the following loss function:

$$\mathcal{L} = \sum_{u_i \in \mathcal{U}^o} ||f(\mathbf{u}_i^s;\phi) - \mathbf{u}_i^t||_2. \tag{3}$$

In real-world CDR scenarios, the overlap of users between the two domains is typically minimal, usually no more than 10%, which poses a challenge for traditional embedding-based methods to train a mapping function that performs well for cross-domain recommendation of non-overlapping users. To this end, we adopt an encoder-decoder framework in both the source and target domains, using autoencoders to extract information from non-overlapping users. They can be formulated as follows:

$$\mathbf{h} = Encoder_s(\mathbf{u}). \tag{4}$$

$$\hat{\mathbf{u}} = Decoder_s(\mathbf{h}). \tag{5}$$

The variable $\mathbf{u}$ denotes the input, which is the user embedding of the source domain. The variable $\mathbf{h}$ denotes our intermediate layer, which exists in a low-dimensional space. The output $\hat{\mathbf{u}}$ represents the reconstruction of the input, i.e., the source domain user embedding. Our goal in this part is to make the output $\hat{\mathbf{u}}$ as close as possible to the input $\mathbf{u}$ and obtain reasonable domain-invariant factors. Therefore, for the source domain, we need to minimize the following reconstruction loss function:

$$\mathcal{L}_{re}^{s} = \sum\nolimits_{u \in \mathcal{U}^{s}} \| \hat{\mathbf{u}} - \mathbf{u} \|_{2}. \tag{6}$$

Similarly, we adopt the same framework for the autoencoder in the target domain as in the source domain. Therefore, for the target domain, we also need to minimize the following reconstruction loss function:

$$\mathcal{L}_{re}^{t} = \sum\nolimits_{u \in \mathcal{U}^{t}} \| \hat{\mathbf{u}} - \mathbf{u} \|_{2}. \tag{7}$$

And we need to minimize the following domain-invariant loss function in this part:

$$\mathcal{L}_{domain-invariant} = \sum\nolimits_{u_i \in \mathcal{U}^{o}} ||Encoder_{s}(\mathbf{u}_{i}^{s}) - Encoder_{t}(\mathbf{u}_{i}^{t})||_{2}. \tag{8}$$

3.4 Task-based Optimization

The objective of task-based optimization is to optimize the transformed user embedding based on the task. Traditional embedding mapping methods mainly use matrix factorization or fully connected neural network models to map the source domain user embedding to the transformed target domain user embedding.

This method has demonstrated good performance in most models, but several issues exist. Unreasonable user embeddings and losing a large amount of rating information lead to the decrease in model accuracy. To tackle these issues, we propose a task-oriented optimization method to train the autoencoder network for the source and target domains. This optimization process directly utilizes the actual target domain users' item ratings as the optimization objective during training. Therefore, in this part, we aim to minimize the following task-oriented loss function:

$$\mathcal{L}_{task-oriented} = \frac{1}{|\mathcal{R}_{o}^{t}|} \sum\nolimits_{r_{ij} \in \mathcal{R}_{o}^{t}} \left(r_{ij} - g(\mathbf{u}_{i}^{s}; \theta)\boldsymbol{v}_{j}^{t}\right)^{2}. \tag{9}$$

where $\mathcal{R}_{o}^{t} = \{r_{ij} | u_i \in \boldsymbol{\mathcal{U}^{o}}, v_j \in \boldsymbol{\mathcal{V}^{t}}\}$ denotes the interactions of overlapping users in the target domain, g denotes the function process that transforms the source domain user embedding using the autoencoder network framework in both the source and target domains, as previously mentioned. g is parameterized by θ. $|\mathcal{R}_{o}^{t}|$ is the total number of user ratings for items in the target domain.

3.5 Joint Training

Initially, it is essential to delineate the joint training tasks required in this study, which include the four tasks. The four tasks are respectively source domain autoencoder training, target domain autoencoder training, domain-invariant factor extraction, and optimization based on real rating data. To execute multitasking concurrently, joint training for the aforementioned four tasks is conducted. More specifically, the source and target domain autoencoder models are trained using all users from both domains. Additionally, the third and fourth tasks are trained with overlapping users from both domains, while the remaining overlapping users serve as the testing set. Consequently, according to formula 6、7、8 and 9, the following joint training loss should be minimized:

$$\mathcal{L} = \mathcal{L}_{re}^{s} + \mathcal{L}_{re}^{t} + \gamma * \mathcal{L}_{domain-invariant} + \delta * \mathcal{L}_{task-oriented}. \tag{10}$$

where γ and δ represent hyperparameters employed to adjust relative weights.

Contrasting with most existing mainstream methods, which adhere to the embedding mapping scheme and train a single model for each task sequentially, our approach jointly trains four tasks. These tasks consist of two autoencoder tasks, a dimensionality reduction task for extracting domain-invariant factors, and a task optimized based on real rating data. These four tasks mutually regulate, share parameters, and utilize implicit data augmentation, thereby enhancing the model's performance.

4 EXPERIMENTS

We conduct experiments to answer the following research questions: (Q1) How does our proposed CDR model perform in the cold-start cross-domain recommendation task compared to other baseline models that address this issue? (Q2) Does the use of non-overlapping user information and task-based optimization strategies help our model? (Q3) Why does our model achieve better performance?

4.1 Experiment Setup

Datasets. In order to conduct experiments directly in real CDR scenarios, we used the Amazon review dataset, which follows the approach of most existing methods[1, 3, 4, 9-11]. The specific statistics are shown in Table 1. Specifically, we proposed two real CDR tasks using the Amazon-5cores dataset, in which each user and item has at least five ratings. To better align with real-world scenarios, we used all the data in the dataset for CDR experiments.

Evaluation Metrics. The ground truth ratings in the Amazon review dataset are integers from 0 to 5. Following [3, 9, 11], we used the mean absolute error (MAE) and root mean square error (RMSE) as standards, which are commonly used metrics in regression tasks.

Baselines.

TGT: Only applies the MF model in the target domain and trains it with the target domain data.

CMF[6] : An enhanced model that shares the embedding of overlapping users between the source and target domains, without considering the differences between the domains.

EMCDR[3]: A commonly used CDR model that mainly learns a mapping function as a bridge between the source and target domains for user embeddings.

SSCDR[2]: Uses a semi-supervised approach and adds distance information encoded according to unlabeled data in learning the mapping function.

DCDCSR[4]: Also a bridge-based method that considers the sparsity of individual user and item ratings in different domains in learning the mapping function.

PTUPCDR[11]: Considers the differences in user interests and adds an attention mechanism to the learning of the source domain user embeddings, learning a personalized bridge function.

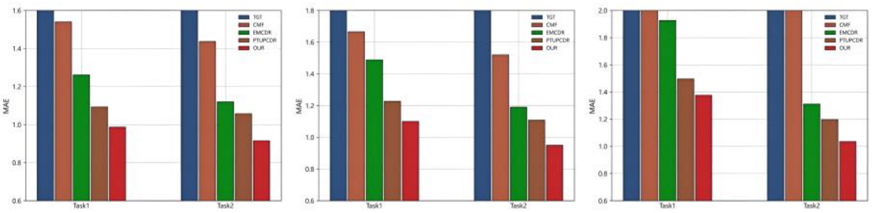
Implementation Details. For optimization, we utilized the Adam optimizer[12] for all tasks and models. Following[3], we randomly selected a portion of overlapping users to construct the test set, with the proportion set to β, while the remaining users were allocated to the training set for the mapping function. To achieve reliable results, we ran the code randomly five times and obtained the average performance, as shown in Figure 2 and Table 2.

Table 2. Cold-start results (RMSE) of two cross-domain tasks. We report the mean results over five runs. The best and the second best results are highlighted by boldface and underlined respectively. Δ% denotes relative improvement over the best baseline.

Task	Task1			Task2		
Method \ β	10%	50%	90%	10%	50%	90%
TGT	5.097	5.158	5.177	4.798	4.769	4.824
CMF	2.029	2.212	2.929	1.883	1.987	3.338
DCDCSR	1.913	2.343	3.206	1.734	2.055	2.770
SSCDR	1.632	1.921	2.432	1.652	1.560	1.702
EMCDR	1.566	1.846	2.319	1.408	1.503	1.664
PTUPCDR	<u>1.424</u>	<u>1.648</u>	<u>2.059</u>	<u>1.364</u>	<u>1.443</u>	<u>1.587</u>
DITOCDR	**1.320**	**1.527**	**1.947**	**1.212**	**1.275**	**1.429**
DITOCDR_a	1.501	1.734	2.211	1.345	1.432	1.588
DITOCDR_b	1.438	1.653	2.142	1.311	1.416	1.546
DITOCDR_c	1.572	1.853	2.302	1.397	1.514	1.652
Δ%	7.30	7.34	5.43	11.14	11.64	9.95

4.2 Overall Performance of DITOCDR（Q1）

In this section, we discuss the experimental results of DITOCDR model in cold-start CDR scenarios. Based on existing bridge-based methods[1-4, 9-11], the performance of our DITOCDR model in cold-start CDR is presented in Table 2 and Figure 2 for two real-world scenarios. The best performance in the experimental results is highlighted in bold. Δ% denotes the relative improvement of our model compared to the previous best baseline. Further analysis reveals the following two findings: (1) CMF is a data augmentation model that merges data from the source and target domains, improving the cold-start problem by employing auxiliary data for recommendations in the target domain. However, the performance of most baseline models surpasses that of CMF. The crux of the issue is that CMF disregards the differences between data from the source and target domains, potentially resulting in domain shift. Conversely, other baseline models are bridge-based methods that learn mapping functions and effectively alleviate this issue, which impedes CDR. (2) Lastly, we observe that DITOCDR's performance consistently outperforms the best baseline across all scenarios, indicating the efficacy of our proposed DITOCDR model in addressing the cold-start recommendation problem.



(a) β = 10% (b) β = 50% (c) β = 90%

Fig. 2. Cold-start experiments on TGT, CMF, EMCDR, PTUPCDR and DITOCDR for different proportions of test (cold-start) users : (a) β = 10%, (b) β = 50%, and (c) β = 90%.

4.3 Ablation study（Q2）

Table 2 shows the results of DITOCDR_a, an ablation study we conducted to investigate the impact of non-overlapping user information on our model. In this experiment, we only used overlapping user data to train the autoencoder between the two domains, instead of using all user information from the source and target domains to train their respective autoencoder frameworks. The results indicate that using only the overlapping user information and adopting a task-based optimization strategy to implement the entire model significantly decreases the recommended performance of the model. Although its performance is weaker than the PTUPCDR model, it is still slightly stronger than the EMCDCR model. These findings suggest that introducing more auxiliary information, such as non-overlapping user information, can improve the recommendation accuracy of cross-domain recommendation. This is particularly important because it can more effectively alleviate the cold start problem that is commonly encountered in CDR.

We also conducted DITOCDR_b, an ablation study to investigate the impact of task-based optimization strategy on our model. In this experiment, we directly learned the target domain user embeddings learned in the latent factor model to predict user interests in target domain items for CDR, instead of learning the target domain user embeddings through a task-based optimization strategy. The data in Table 2 show that when we use non-overlapping user information for CDR without adopting a task-based optimization strategy, the recommended performance of the model still significantly decreases. Although its performance is weaker than the PTUPCDR model, it is still stronger than the EMCDR model. These findings suggest that task-based optimization strategy is also helpful for our model, as it directly learns the ground truth rating labels, which can lead to more reasonable embeddings for users compared to previous models that only learned user embeddings.

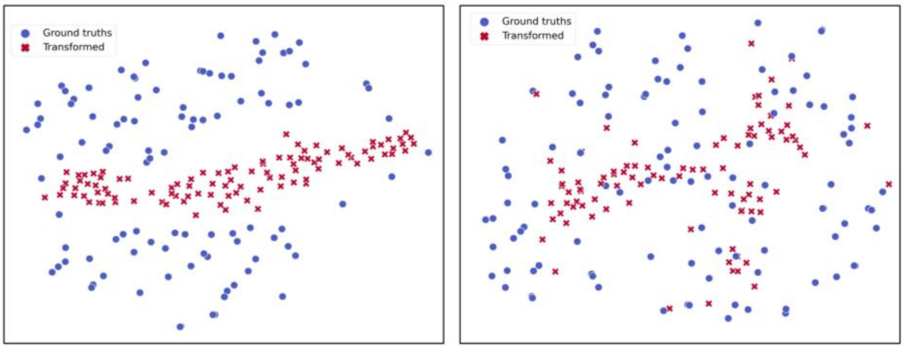
Finally, we conducted DITOCDR_c, an ablation study without using non-overlapping user information and a task-based optimization strategy. The performance of this experiment decreased the most compared to DITOCDR, and is very close to the EMCDR model's performance. This further illustrates the critical importance of non-overlapping user information for training the autoencoder framework of our model in the source and target domains, and the usefulness of task-based optimization strategy for better learning the target embeddings, i.e., the process of transforming user embeddings from the source domain to the target domain. Overall, both non-overlapping user information and task-based optimization strategy are crucial for improving the performance of our model for CDR.

4.4 Explanation of the Improvement（Q3）

In this section, we conducted experiments to visualize potential factors and illustrate why our DITOCDR model has further improvements. Specifically, we analyzed the embeddings in the feature space of the target domain to investigate the superiority of DITOCDR over EMCDR.

Following the default settings of t-SNE[13], We used it in Scikit-learn to visualize the user embeddings learned by EMCDR and DITOCDR on Task 2, with β = 0.1. Figure 3(a) shows the user embeddings learned by EMCDR, while Figure 3(b) shows the visualized embeddings learned by DITOCDR. The blue dots denote the target embeddings of the target domain, which are considered the ground truth, and the red cross points denote the transformed embeddings learned by our model. To ensure clarity, we randomly selected 150 users for plotting.

Ideally, the distribution of the user embeddings we learned should be roughly similar to the distribution of the target embeddings. From Figure 3(a), we observe that the distribution of the actual user embeddings is scattered, but the embeddings learned by EMCDR are relatively concentrated. This is due to the fact that EMCDR learns from overlapping user data, while in practice, the majority of users in most CDR tasks do not overlap. Therefore, ignoring the information of most non-overlapping users leads to a significant accuracy loss. Additionally, EMCDR directly learns the mapping function from the source domain user embeddings to the target domain user embeddings, while the target domain embeddings are learned through latent factor models, which leads to inaccurate target domain embeddings and a further performance decline. In contrast, DITOCDR achieves better results, as shown in Figure 3(b). The transformed user embeddings learned by DITOCDR are not clustered together like those learned by EMCDR. Instead, they are scattered across the feature space of the target domain, indicating the superiority of using non-overlapping user data and task-based optimization strategies. Moreover, the distribution of the transformed embeddings by DITOCDR fits the distribution of the target embeddings better, which may be the fundamental reason why DITOCDR can achieve better overall performance.



(a) EMCDR (b) DITOCDR

Fig. 3. t-SNE visualization of randomly sampled user ground truths embeddings in target-domain feature space and transformed user embeddings. (a) and (b) denote the visualization results of EMCDR and DITOCDR, respectively.

5 CONCLUSION

In this paper, we studied the cold-start problem in cross-domain recommendation (CDR). Many popular CDR methods assume overlap between users and directly learn target embeddings. However, disregarding information from non-overlapping users and learning unreasonable target embeddings can lead to challenges in model generalization. To overcome these issues, we propose a new framework called Domain-Invariant Task Optimization for Cross-domain Recommendation (DITOCDR). Briefly, two autoencoder networks learn all user data from source and target domains and extract domain-invariant factors. The training of target embeddings is optimized using task-based optimization strategies. We conducted extensive experiments on three real-world datasets from Amazon to evaluate DITOCDR. Experimental results demonstrate the excellent performance and effectiveness of our proposed DITOCDR.

References

1. Fu, W., Peng, Z., Wang, S., Xu, Y., Li, J.: Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In: AAAI (2019).
2. Kang, S., Hwang, J., Lee, D., Yu, H.: Semi-supervised learning for cross-domain recommendation to cold-start users. In: CIKM (2019).
3. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: An embedding and mapping approach. In: IJCAI (2017).
4. Zhu, F., Wang, Y., Chen, C., Liu, G., Orgun, M., Wu, J.: A deep framework for cross-domain and cross-system recommendations. arXiv preprint arXiv:2009.06215 (2020).
5. Wang, T., Zhuang, F., Zhang, Z., Wang, D., Zhou, J., He, Q.: Low-dimensional alignment for cross-domain recommendation. In: CIKM (2021).
6. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: SIGKDD (2008).
7. Pan, W., Xiang, E., Liu, N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: AAAI (2010).
8. Zhang, Y., Liu, Y., Han, P., Miao, C., Cui, L., Li, B., Tang, H.: Learning Personalized Itemset Mapping for Cross-Domain Recommendation. In: IJCAI (2020).
9. Zhao, C., Li, C., Xiao, R., Deng, H., Sun, A.: CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In: SIGIR (2020).
10. Zhu, Y., Ge, K., Zhuang, F., Xie, R., Xi, D., Zhang, X., Lin, L., He, Q.: Transfer-meta framework for cross-domain recommendation to cold-start users. In: SIGIR (2021).
11. Zhu, Y., Tang, Z., Liu, Y., Zhuang, F., Xie, R., Zhang, X., Lin, L., He, Q.: Personalized transfer of user preferences for cross-domain recommendation. In: WSDM (2022).
12. Kingma, D.P., J, Ba.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
13. Donahue, J., Jia, Y., Vinyals, J., Hoffman, J., Zhang, N.: DeCAF: A deep convolutional activation feature for generic visual recognition. In: ICML (2014).