A Graph Neural Network for Cross-domain Recommendation Based on Transfer and Inter-domain Contrastive Learning

Caihong Mu[1], Jiahui Ying[1], Yunfei Fang[1](✉), and Yi Liu[2]

[1] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Collaborative Innovation Center of Quantum Information of Shaanxi Province, School of Artificial Intelligence, Xidian University, Xi'an, China
caihongm@mail.xidian.edu.cn, {jhying, fangyunfeixd}@stu.xidian.edu.cn

[2] School of Electronic Engineering, Xidian University, Xi'an, China
yiliu@xidian.edu.cn

Abstract. Cross-domain recommendation (CDR) is an effective method to deal with the problem of data sparsity in recommender systems. However, most of the existing CDR methods belong to single-target CDR, which only improve the recommendation effect of the target domain without considering the effect of the source domain. Meanwhile, the existing dual-target or multi-target CDR methods do not consider the differences between domains during the feature transfer. To address these problems, this paper proposes a graph neural network for CDR based on transfer and inter-domain contrastive learning (TCLCDR). Firstly, user-item graphs of two domains are constructed, and data from both domains are used to alleviate the problem of data sparsity. Secondly, a graph convolutional transfer layer is introduced to make the information of the two domains transfer bidirectionally and alleviate the problem of negative transfer. Finally, contrastive learning is performed on the overlapping users or items in the two domains, and the self-supervised contrastive learning task and supervised learning task are jointly trained to alleviate the differences between the two domain.

Keywords: Cross-domain recommendation · User-item graphs · Transfer · Contrastive learning

1 Introduction

Recommendation algorithms are usually divided into three categories: content-based [1], collaborative filtering-based [2] and hybrid ones [3]. Cross-domain recommendation (CDR) algorithm is a kind of model-based collaborative filtering recommendation algorithm, which is a challenge in recommender systems.

To further improve the model's ability to extract representation vectors and the recommendation performance of the model in two domains, this paper proposes a graph neural

Z. Jin et al. (Eds.): KSEM 2023, LNAI 14119, pp. 226–234, 2023.
https://doi.org/10.1007/978-3-031-40289-0_18

network for CDR based on transfer and inter-domain contrastive learning (TCLCDR). The main contributions of this paper are as follows:

- The information of two domains is used to alleviate the problem of data sparsity.
- A graph convolutional transfer layer (GCTL) is designed to make full use of the information of its own domain and the other domain, which improves the ability of the model to extract representation vectors and alleviates the problem of negative transfer.
- Considering that the similarity of overlapping users or items in the two domains is greater than that of non-overlapping users or items, a contrastive learning loss function (CLLF) is proposed to alleviate the difference between the two domains during information transfer.

2 Related Work

Single-domain Recommendation. In 2018, Berg et al. [4] proposed the Graph convolutional matrix completion for bipartite edge prediction (GCMC),which effectively combined user interaction data and side information to predict the score. In 2019, Wang et al. [5] proposed Neural Graph Collaborative Filtering (NGCF). In 2020, He et al. [6] proposed a Simplifying and Powering Graph Convolution Network for Recommendation (LightGCN). Compared with NGCF, LightGCN simplified feature transformation and nonlinear activation, which improved the recommendation effect while reducing model training time.

Single-target CDR. The task goal of the single-target CDR is to use the data-rich source domain for modeling to improve the recommendation accuracy of the model for the data-sparse target domain.

Dual-target CDR. In 2018, Hu et al. [7] proposed Collaborative Cross Networks for CDR (CoNet). The algorithm cross-mapped and connected the hidden layers of the two domains to form a collaborative cross network. In 2019, Zhao et al. [8] proposed a CDR via Preference Propagation GraphNet (PPGN). The algorithm put the users and items of the two domains into a graph, and then aggregated the information of the two domains through graph convolution. In 2020, Liu et al. [9] proposed a CDR via Bi-directional Transfer Graph Collaborative Filtering Networks (BiTGCF). The bidirectional transfer learning method was used to realize the mutual transfer of knowledge between the two domains.

Multi-target CDR. The task goal of the multi-target CDR is to improve the recommendation accuracy of all domains by using data from multiple domains.

3 Method

This section will describe the TCLCDR's framework (see Fig, 1). Firstly, the model uses the user-item rating matrixes of the two domains to construct the source and the target domain user-item graph respectively, and aggregates the information of the two domains through the GCTL. Specifically, taking the update of the source-domain user representation vector (URV) as an example, it can be implemented in three steps: (1) the user-item graphs of the two domains are input into the graph convolutional layer (GCL) to obtain the user graph convolutional representation vectors (GCRVs); (2) The user GCRV in the source domain and that in the target domain are transmitted to the transfer layer to obtain the user transfer representation vector (TRV). (3) Then, the source-domain user GCRV and the user TRV are aggregated to obtain a new source-domain URV. The other representation vectors are updated in the same way. Finally, the representation vector is transmitted to the prediction layer to output the recommendation list. Considering that the representation vectors of overlapping users or items should be more similar than those of non-overlapping users or items, the model performs contrastive learning in the source and target domain, and jointly trains the self-supervised contrastive learning task and the supervised learning task.

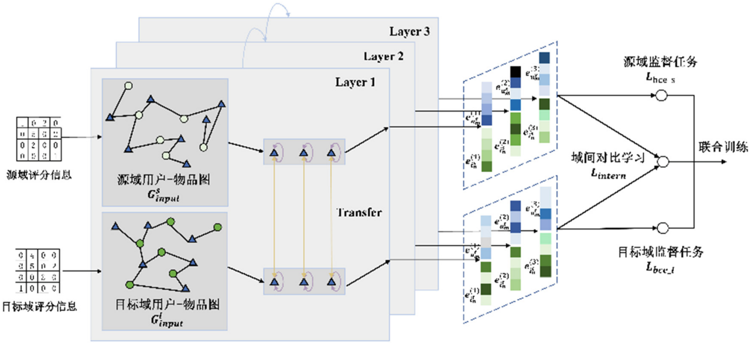


Fig. 1. The overall framework of TCLCDR

3.1 Construction of Graphs

Taking M^s users $u^s = u_1^s, u_2^s, \ldots, u_{M^s}^s$ and N^s items $i^s = i_1^s, i_2^s, \ldots, i_{N^s}^s$ in the source domain as nodes and all interaction information L^s as edges, a user-item graph in the source domain is constructed in the form of a matrix as shown in Eq. (1):

$$G_{input}^s = \begin{bmatrix} 0^{M^s \times M^s} & R^s \\ R^{s^T} & 0^{N^s \times N^s} \end{bmatrix} \tag{1}$$

where R^s denotes the matrix of size $M^s \times N^s$, the m-th row and n-th column array element is $R_{m,n}^s$, and R^{s^T} denotes the transpose result of R^s. if iteraction exists between u_m^s and i_n^s, $R_{m,n}^s = 1$; otherwise, $R_{m,n}^s = 0$. The target-domain user-item graph is constructed in the same way.

3.2 Graph Convolutional Transfer Layer

The representation vector $\{e_{u^s}^{(0)}, e_{i^s}^{(0)}\}$ of users and items in the source domain and the representation vector $\{e_{u^t}^{(0)}, e_{i^t}^{(0)}\}$ of users and items in the target domain are initialized.

Extraction of the Graph Convolutional Representation Vector. The source-domain user-item graph is input into the GCL to obtain the source-domain user and item GCRV $\{\left(e_{u_m^s}^{gcf}\right)^{(k)}, \left(e_{i_n^s}^{gcf}\right)^{(k)}\}$, and the target-domain user-item graph is input into the GCL to obtain the target-domain user and item GCRV $\{\left(e_{u_m^t}^{gcf}\right)^{(k)}, \left(e_{i_n^t}^{gcf}\right)^{(k)}\}$. The source-domain user GCRV is extracted as shown in Eq. (2):

$$\left(e_{u_m^s}^{gcf}\right)^{(k)} = \left(e_{u_m^s}^{gcf}\right)^{(k\text{-}1)} + \frac{\sum_{n\in G_m^s} G_{m,n}^s \left(e_{i_n^s}^{gcf}\right)^{(k\text{-}1)}}{\sum_{n=0}^{M^s+N^s-1} G_{m,n}^s} \tag{2}$$

where k is the number of GCL, $\left(e_{u_m^s}^{gcf}\right)^{(k)} \left(e_{i_n^s}^{gcf}\right)^{(k)}$ is the k-th GCRV, $G_m^s = \{n|G_{m,n}^s > 0\}$ is the collection of items that u_m^s interacts with in the source-domain. The other GCRV are calculated in the same way.

Extraction of the Transfer Representation Vector. The transfer layer is an important part of the TCLCDR. By extracting the TRV, the model uses the information of its own domain and other domains, and realizes the information transfer between the two domains. Specifically, the TRV can be obtained by transferring the source-domain GCRV and the target-domain GCRV to the transfer layer, as shown in Eq. (3) and (4):

$$(e_{u_m}^{tr})^{(k)} = W_u(l_{u_m^s}(e_{u_m^s}^{gcf})^{(k)} + l_{u_m^t}(e_{u_m^t}^{gcf})^{(k)} \tag{3}$$

$$(e_{i_n}^{tr})^{(k)} = W_i(l_{i_n^s}(e_{i_n^s}^{gcf})^{(k)} + l_{i_n^t}(e_{i_n^t}^{gcf})^{(k)}) \tag{4}$$

where W_u, W_i represent the mapping matrix, and $l_{u_m^s}$ is calculated as shown in Eq. (5):

$$l_{u_m^s} = \frac{N_{u_m^s}}{N_{u_m^s} + N_{u_m^t}} \tag{5}$$

where $N_{u_m^s}$ and $N_{u_m^t}$ represents the number of interactions of user u_m in the source and target domain, respectively. $l_{u_m^t}$, $l_{i_n^s}$ and $l_{i_n^t}$ are calculated in the same way.

Aggregataion of the Representation Vectors. Finally, the TRV and the GCRV are aggregated to update the representation vectors of the source and target domain. The aggregation of the URVs is shown in Eq. (6) and (7):

$$e_{u_m^s}^{(k)} = (e_{u_m}^{tr})^{(k)} + \lambda^s(e_{u_m^s}^{gcf})^{(k)} + (1\text{-}\lambda^s)(e_{u_m^t}^{gcf})^{(k)} \tag{6}$$

$$e_{u_m^t}^{(k)} = (e_{u_m}^{tr})^{(k)} + \lambda^t(e_{u_m^t}^{gcf})^{(k)} + (1\text{-}\lambda^t)(e_{u_m^s}^{gcf})^{(k)} \tag{7}$$

where λ^s, λ^t represent hyperparameters ranging from 0 to 1 that control the weights of the graph convolutional vectors for the source and target domain. The same goes for item representation vectors (IRVs).

After obtaining the representation vector of each GCTL, the final URV and IRV are obtained by concatenation. When the number of layers is 3, the user representations in the source domain are obtained as shown in Eq. (8):

$$e_{u_m^s} = \text{concat}\left(e_{u_m^s}^{(1)}, e_{u_m^s}^{(2)}, e_{u_m^s}^{(3)}\right) \tag{8}$$

3.3 Construction of the Contrastive Learning Loss Function

From the user's perspective, each user u_m has two different representation vectors $e_{u_m^s}$, $e_{u_m^t}$ in the source and target domains after passing through the GCTL. Although the same user may have different preferences and interaction histories in the two domains, the vectors of its source domain and its target domain should be more similar than the representation vectors of other users in the target domain, so the loss function of user-based inter-domain contrastive learning is constructed as shown in Eq. (9):

$$L_{inter}^{u} = \sum_{u_m^s \in u^s} \text{-log} \frac{\exp\left(s\left(e_{u_m^s}, e_{u_m^t}\right)/\tau\right)}{\sum_{u_{m'} \in u^t} \exp\left(s\left(e_{u_m^s}, e_{u_{m'}^t}\right)/\tau\right)} \tag{9}$$

where $s(\cdot)$ is the similarity function (cosine similarity is used in this paper), and τ is the temperature parameter. The loss function of item-based inter-domain contrastive learning (L_{inter}^{i}) is calculated in the same way.

3.4 Rating Prediction and Model Training

After obtaining the URV and IRV, the user's rating of the item is calculated through the prediction layer, and the calculation method is shown in Eq. (10):

$$\hat{y}_{u_m i_n}^{s} = < e_{u_m}^{s}, e_{i_n}^{s} > \tag{10}$$

The loss function of the supervised learning task uses the cross-entropy loss function. Because the effect of the two domains needs to be improved simultaneously, the loss function of the two domains needs to be calculated, as shown in Eq. (11) and (12):

$$L_{bce_s} = -\sum_{(u_m^s, i_n^s \in O^s)} y_{u_m i_n}^{s} \log\left(\hat{y}_{u_m i_n}^{s}\right) + \left(1\text{-}y_{u_m i_n}^{s}\right)\log\left(1\text{-}\hat{y}_{u_m i_n}^{s}\right) \tag{11}$$

$$L_{bce_t} = -\sum_{(u_m^t, i_n^t \in O^t)} y_{u_m i_n}^{t} \log(\hat{y}_{u_m i_n}^{t}) + (1\text{-}y_{u_m i_n}^{t})\log(1\text{-}\hat{y}_{u_m i_n}^{t}) \tag{12}$$

where O^s represents the training sample set composed of source-domain users and items, $y_{u_m i_n}^{s}$ represents the real label of the source-domain training set. When u_m^s interacts with i_n^s, $y_{u_m i_n}^{s} = 1$; otherwise, $y_{u_m i_n}^{s} = 0$. The same goes for the target domain.

Finally, contrastive learning loss functions and cross-entropy loss functions are combined to jointly train the TCLCDR, and the final loss function is shown in Eq. (13):

$$L = L_{bce_s} + L_{bce_t} + \lambda_1 (L_{inter}^{u} + L_{inter}^{i}) \tag{13}$$

where λ_1 and λ_2 represent hyperparameters ranging from 0 to 1.

4 Experiments and Results

4.1 Datasets

In this paper, we conduct experiments on the Amazon dataset including: "*Electronics* (*Elec*)", "*Cell Phones* (*Cell*)", "*Sports and Outdoors* (*Sport*)", "*Clothing Shoes and Jewelry* (*Cloth*)", "*Grocery and Gourmet Food* (*Groc*)", "*Tools and Home* (*Tool*)". For each dataset, We filter the data to include users who have at least 5 interactions and items that have at least 10 interactions [5]. These six datasets are then used to form three groups of datasets for experiments. Finally, we keep the overlapping users and all items. The statistics of datasets are shown in Table 1.

Table 1. The Statistics of Datasets

dataset	users	items	interactions	Sparsity (%)
Elec & Cell	3325&3325	39463&18462	118879&53732	99.90&99.91
Sport & Cloth	9928&9928	32310&41303	102540&97757	99.96&99.97
Groc & Tool	22746&22746	49217&66961	340857&333588	99.97&99.98

We use the leave-one-out to build the training and test sets. The leave-one-out is different from K-fold verification. The leave-one-out takes the last interaction item of the user as the test set of the user, and the remaining interaction items as the training set.

4.2 Baseline Model Comparison

In this section, we compare TCLCDR with three single-domain models and three CDR models on three datasets (*Elec & Cell, Sport & Cloth* and *Groc & Tool*) by using two metrics including Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG). The larger the values of these two metrics, the better the performance of the model. The results are shown in Table 2, where the best results are in bold.

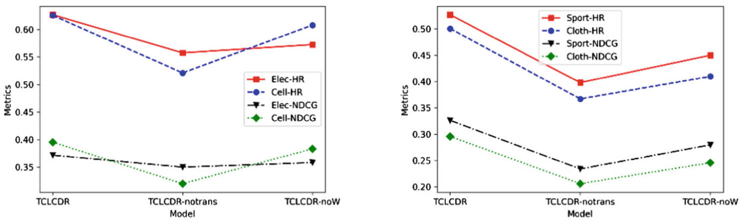
It can be seen from Table 2 that two metrics values obtained by TCLCDR on all the datasets are the best, which illustrates the effectiveness of TCLCDR. For the single-domain model, the indicators of the LightGCN are always higher than those of NGCF and GCMC, indicating that the simplified feature transformation of LightGCNF can improve the recommendation effect. For the cross-domain model BiTGCF, the performance of which is greatly improved compared with other baseline models, indicating that the bidirectional transfer effect in BiTGCF is significant. On the whole, the proposed TCLCDR outperforms other baseline models significantly.

4.3 Model Ablation Experiments

The Effect of Transfer Layers. Three models are compared, which are (1) TCLCDR, (2) TCLCDR-notrans without using GCTL, and (3) TCLCDR-noW without the mapping matrix in the GCTL. Figure 2 shows the experimental results.

Table 2. Baseline Model Comparison

dataset	metrics	Single-Domain Model			Cross-Domain Model			
		GCMC	NGCF	LightGCN	CoNet	PPGN	BiTGCF	TCLCDR
Elec	HR	0.3883	0.4096	0.4087	0.4484	0.4600	0.5657	**0.6364**
	NDCG	0.2238	0.2548	0.2560	0.2861	0.2644	0.3535	**0.3793**
Cell	HR	0.4063	0.4334	0.4499	0.4643	0.5126	0.5621	**0.6496**
	NDCG	0.2364	0.2749	0.2847	0.3004	0.2504	0.3542	**0.4044**
Sport	HR	0.3405	0.3360	0.3917	0.3667	0.3659	0.4473	**0.5295**
	NDCG	0.1874	0.1953	0.2470	0.2244	0.1512	0.2768	**0.3280**
Cloth	HR	0.2878	0.2953	0.3219	0.3235	0.3380	0.4191	**0.5038**
	NDCG	0.1581	0.1633	0.1931	0.2127	0.1445	0.2474	**0.2961**
Groc	HR	0.4808	0.4859	0.5528	0.5099	0.5555	0.5683	**0.6418**
	NDCG	0.3011	0.3050	0.3584	0.3173	0.2867	0.3935	**0.4527**
Tool	HR	0.4268	0.4334	0.4836	0.5626	0.5079	0.4956	**0.5720**
	NDCG	0.2438	0.2506	0.2872	0.3696	0.2837	0.3203	**0.3812**



(a) Metrics on *Elec & Cell* (b) Metrics on *Sport & Cloth*

Fig. 2. Experimental results for different transfer layers

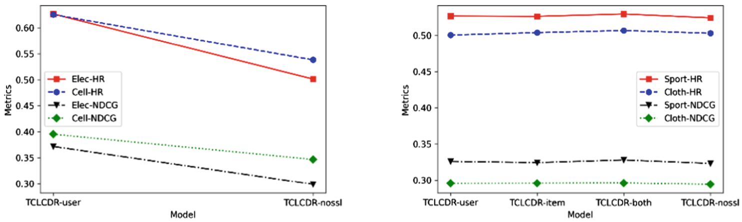
From Fig. 2, it can be seen that on the two datasets, when the model uses GCTL, the values of two metrics in the two domains are higher than those when GCTL is not used or the mapping matrix is not used.

The Effect of Contrastive Learning Loss Function. We design four variant models based on TCLCDR to test the effect of CLLF as shown in Table 3, where 'nossl' denotes the model not using self-supervised contrastive learning. Figure 3 shows the results.

From Fig. 3, it can be seen that on the dataset *Elec & Cell*, when the model uses L^u_{inter}, the values of two metrics in the two domains are higher than those when L^u_{inter} is not used. On the dataset *Sport & Cloth*, in most cases, the values of metrics obtained by the the model using CLLF are higher than those obtained by the one without using CLLF.

Table 3. Models Using Different Contrastive Learning Loss Functions

dataset	*Elec & Cell*		*Sport & Cloth*			
Models	user	nossl	user	item	both	nossl
L_{inter}^{u}	√		√		√	
L_{inter}^{i}				√	√	



(a) Metrics on *Elec & Cell* (b) Metrics on *Sport & Cloth*

Fig. 3. Experimental results for different contrastive learning loss functions

5 Conclusion

A graph neural network for cross-domain recommendation based on transfer and inter-domain contrastive learning (TCLCDR) is proposed to recommend a list of favorite items to users in two domains. Compared with the baseline models, the extensive experimental results show that the proposed TCLCDR performs better in the hit rate and normalized discounted cumulative gain. The results of ablation experiments show that the graph convolutional transfer layer and contrastive learning can improve the model's ability to extract representation vectors and help the model generate more accurate item recommendation lists in both domains.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (Nos. 62077038, 61672405, 62176196 and 62271374).

References

1. Balabanovi, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)
2. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Conference on Computer Supported Cooperative Work, pp. 241–250 (2000)
3. Chen, W., Niu, Z., Zhao, X., Li, Y.: A hybrid recommendation algorithm adapted in e-learning environments. World Wide Web **17**(2), 271–284 (2012). https://doi.org/10.1007/s11280-012-0187-z
4. Wu, Y., Liu, H., Yang, Y.: Graph convolutional matrix completion for bipartite edge prediction. In: International Conference on Knowledge Discovery and Information Retrieval, pp. 49–58 (2018)

5. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: 42nd International ACM SIGIR Conference, pp. 165–174 (2019)
6. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M: Lightgcn: simplifying and powering graph convolution network for recommendation. In: 43rd International ACM SIGIR Conference, pp. 639–648 (2020)
7. Hu, G., Zhang, Y., Yang, Q.: Conet: collaborative cross networks for cross-domain recommendation. In: 27th ACM International Conference on Information and Knowledge Management, pp. 667–676 (2014)
8. Zhao, C., Li, C., Fu, C.: Cross-domain recommendation via preference propagation graphnet. In: 28th ACM International Conference on Information and Knowledge Management, pp. 2165–2168 (2019)
9. Liu, M., Li, J., Li, G., Pan, P.: Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In: 29th ACM International Conference on Information & Knowledge Management, pp. 885–894 (2020)