“Beyond the Overlapping Users: Cross-Domain Recommendation via Adaptive Anchor Link Learning” Reproduction

Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, Jibing Gong

Abstract

Cross domain recommendation (CDR) improves recommendation performance by combining auxiliary information from multiple domains. Existing CDR methods mainly rely on overlapping users to transfer preference information between source domain identities and target domain identities belonging to the same user. However, this heuristic assumption is not applicable in a large number of practical application scenarios. On the one hand, in practical recommendation scenarios, it is difficult to obtain the interaction information of the same user in different domains. On the other hand, individual users may exhibit different or even conflicting preferences in different fields, leading to potential noise. In this article, authors consider the anchor links between users in different domains as learnable parameters to learn cross domain correlations related to tasks. A new ALCDR model based on optimal transmission is further proposed, which accurately infers anchor links and deeply aggregates collaborative signals from both intra-domain and inter-domain perspectives. Extensive evaluations have been conducted on five real-world datasets, and experimental results have demonstrated its superiority.

Keywords: Cross-domain recommendation, Optimal transportation, Graph neural networks.

1 Introduction

To address the problem of information overload [5] , recommendation systems (RS) have emerged as a solution, finding widespread application across various real-world domains like online commerce. At the core of these recommendation systems lies the principle of collaborative filtering (CF) [10], which involves the precise modeling of user preferences based on their past interactions with items, encompassing actions like purchases and clicks. Subsequently, recommendations are formulated by leveraging the acquired insights into individual interests. However, the primary hurdle faced by most recommendation systems is the persistent issue of data sparsity, arising due to the limited number of user ratings or feedback provided for significant items.

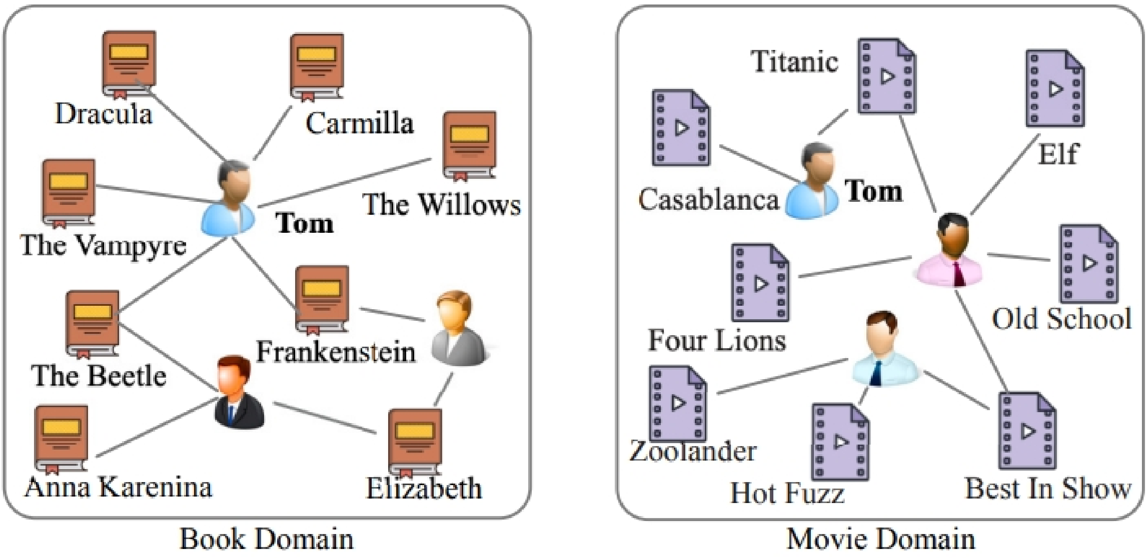

Figure 1. An illustration of the diverse preferences of a single individual over various domains. [13]

Cross domain recommendation has been proven to be a promising strategy that can alleviate the problem of sparsity. The cornerstone of CDR lies in utilizing user item interactions from multiple domains to jointly model user preferences. Existing CDR work typically focuses on two core questions: 1) what to transfer: Mining valuable transferable information from the users in different domains. matrix factorization [6], factorization machines [12], deep neural networks [2], and graph neural networks [1] encodes user historical behavior information to obtain embedded representations of users, thereby achieving learning of user preferences. 2) how to transfer: Designing a effecitve cross-domain konwledge transfer pattern. Cross-domain co-factorization [7] achieves the transfer of user preferences cross domains by designing complex knowledge transfer mechanisms.

In this work, authors aim to delve deeper into the question of ”who to transfer” and go beyond the assumption of ”the same natural person” to reveal the full potential of CDR. Their motivation is to adaptively learn anchor links between identities belonging to different domains, rather than traditional heuristic based static links.

Anchor links should be task related. The recommended lost gradient should be backpropagated to guide the learning of anchor links and establish an end-to-end CDR paradigm;

Anchor links should be sparse. A simple method is to establish anchor links between each pair of source and target identities

Anchor links should support many to many mapping. The target identity may benefit from multiple source identities;

In this work, authors propose a new model ALCDR to jointly learn anchor links and provide recommendations. Formalize cross domain anchor links inference into an optimal transport (OT) problem users from a single domain are considered as a discrete distribution, and optimal transport aims to learn the optimal transport prototype to minimize the cost of transforming one distribution into another. The learned transport prototype is regarded as a signal guiding cross domain information transfer. Their suggestion is very attractive because the OT solver can easily adapt to the suggestion paradigm to facilitate end-to-end CDR. In addition, the approximate optimal prototype implemented by popular OT solvers (such as the Sinkhorn solver) is a sparse plan, which helps to reduce the risk of potential noise. Specifically, ALCDR consists of three main modules: intra-domain aggregation to capture collaborative signals within a single domain, anchor link inference to learn cross domain correlations based on OT minimization, and inter-domain aggregation to deeply fuse valuable knowledge cross domains. From an empirical perspective, ALCDR has been extensively evaluated on five public datasets by myself, and experimental results have demonstrated the superiority of our suggestion. ALCDR main contributions can be summarized in three aspects:

2 Related works

Cross-domain recommendation is an effective method to solve the target domain data sparse and cold start problems with the help of auxiliary (source) domain. Existing CDR methods focus on mining valuable information across domains and how to transfer knowledge across domains. In order to find valuable methods in cross-domain formation, various potential representation learning methods, such as matrix factorization [4], factorization machine [8], deep neural networks [9, 14] and graph neural networks [3], are used in existing CDR methods to capture valuable information from past interactions and encode them into low-dimensional embeddings.

3 Method

3.1 Problem define

Before introducing the methods in the paper, I first provide a problem definition for cross domain recommendation tasks based on previous work [11]. Cross domain recommendation occurs in two domains, one domain is defined as $\mathcal{D} = \{\mathcal{U}, \mathcal{V}, \mathcal{R}\}$ where $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ represents the user set, $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ represents the item set, and matrix $\mathcal{R} \in \mathbb{R}^{m \times n}$ Preserve user-item history interactions. In cross domain recommendation tasks, given input domains $\mathcal{D}^s = \{\mathcal{U}^s, \mathcal{V}^s, \mathcal{R}^s\}$ and $\mathcal{D}^t = \{\mathcal{U}^t, \mathcal{V}^t, \mathcal{R}^t\}$, cross domain recommendation aims to predict items that users may be interested in by utilizing information from both domains. It is worth noting that previous work relied on a subset of overlapping users as the anchor chain connection for $\mathcal{U} = u$. In this study, authors adopted different perspectives, treating anchor links as learning objectives and eliminating the need for pre-collected overlapping users, without relying on the needs of overlapping users.

3.2 Overview

Figure 2 shows their proposed framework, which consists of three key components: intra-domain aggregation, anchor link inference, and inter-domain aggregation. The historical user-item interactions in each domain are modeled as bipartite graphs In the intra-domain aggregation module, graph neural networks (GNNs) are used to capture high-order collaborative signals to derive the required user and item representations. Subsequently, the aggregated user representations are fed to the anchor links inference module to learn cross domain transport prototypes users from the same domain are considered as a discrete distribution, and anchor chain inference can be formulated as the optimal transport problem, which can be effectively solved using the Sinkhorn algorithm. In the inter-domain aggregation module, based on the learned anchor links, user representations are transferred from one domain to another, providing valuable supplementary information. Finally, generate recommendations by calculating the similarity between cross domain user and item representations.

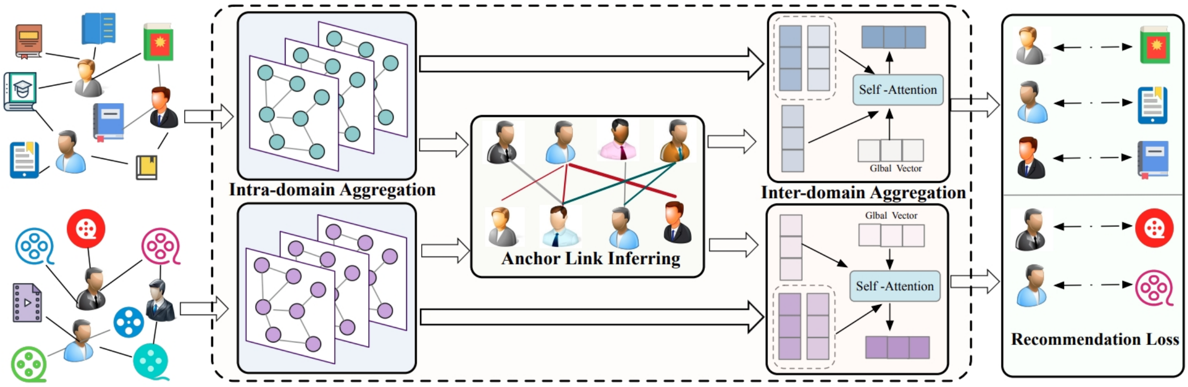

Figure 2. System architecture of ALCDR. [13]

3.3 Intra-domain Aggregation

The core task of a recommendation system is to explore collaboration signals in user-item interaction information. There are already many excellent methods in single domain recommendation tasks that can effectively capture user interests. In recent years, graph convolutional networks have achieved good performance in the field of recommendation to promote user preference modeling. In the ALCDR model, GNN is also used as the backbone network for inter-domain aggregation. The user-item interactions within a single domain is represented as a bipartite graph, where nodes represent users/items and edges represent interaction relationships. A typical graph neural network (GNN) layer updates node representations through the process of neighborhood representation feature transformation, propagation, and aggregation. The update process of the representation of the central node can be formulated as follows in the l-th GNN layer of the middle node L-layer GNN:

$$a_c^{(l)} = AGG(e_n^{(l-1)}, n \in N_c), l \in [L] \tag{1}$$

$$e_c^{(l)} = COMBINE(e_c^{(l-1)}, a_c^{(l)}), l \in [L] \tag{2}$$

where $e_c^{(l)}$ denotes the d-dimensional embedding of node c in l-th layer. AGG refers to the aggregation function to aggregate information from the neighborhood nodes. $COMBINE$ refers to the combination function to update the node representation in the l-th layer based on the embedding $a_c^{(l)}$.

In this work, ALCDR utilizes the LightGCN to aggregate information from the neighborhood nodes. The aggregation function is defined as:

$$e_u^{(l+1)} = \sum_{v \in N_u} \frac{1}{\sqrt{N_u}\sqrt{N_v}} e_v^{(l)} \tag{3}$$

$$e_v^{(l+1)} = \sum_{u \in N_v} \frac{1}{\sqrt{N_v}\sqrt{N_u}} e_u^{(l)} \tag{4}$$

where the $\frac{1}{\sqrt{N_u}\sqrt{N_v}}$ refers to the symmetric normalization term to counteract the potential increase in the scale of embeddings.

The embeddings in the first layer are initialized as $e_u^{(0)}$ and $e_v^{(0)}$, and higher-level embeddings can be iteratively calculated using equations (3) and (4). Afterwards, in the propagation L layers, LightGCN combines the output embeddings from each layer to form the final representation of the user or item:

$$\hat{e}_u = \sum_{l=0}^{L} \alpha_l e_u^{(l)}, \hat{e}_v = \sum_{l=0}^{L} \alpha_l e_v^{(l)} \tag{5}$$

where $\alpha_l \geq 0$ refers to rhe importance of the l-th layer embedding.

Algorithm 1 Optimal transport solver (SolveOT)

Input : cross domain user matrices $\mathrm{U_s} = \{x_i\}_1^n, U_t = \{y_j\}_1^m$

$\sigma = \frac{1}{m} 1_m, T^{(1)} = 1_n 1_m^\top$

$C_{ij} = c(x_i, y_i), A_{ij} = e^{-\frac{c_{ij}}{\beta}}$

for *t = 1,2,3...do* **:**

  $Q = a \odot T^{(t)}$

  for *t = 1,2,3...do* **:**

    $\delta = \frac{1}{nQ\sigma}, \delta = \frac{1}{mQ^\top \delta}$

  $T^{(t+1)} = diag(\delta) Q diag(\sigma)$

return T

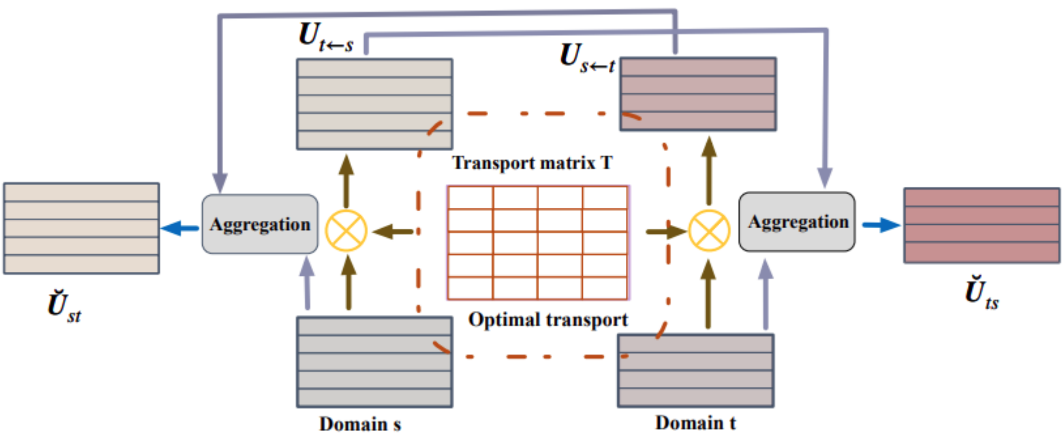

Figure 3. Framework of the anchor link inferring module. [13]

3.4 Anchor Link Inferring

The correlation between cross domain users (identities) is considered a learnable parameter, known as anchor link inference, and is optimized and learned based on downstream recommendation losses Assuming the user embedding matrix of the source domain s is $\mathrm{U_s} \in \mathbb{R}^{m^{(s)} \times d}$ and the user embedding matrix of the target domain t is $\mathrm{U_t}$, where $m^{(s)}$ and $m^{(t)}$ represent the number of users in different domains. Anchor link inference aims to learn the sparse mapping matrix T, where T_{ij} represents the possibility of information from the source domain user $u_i^{(s)}$ being transmitted to the target domain user $u_j^{(t)}$. From a mathematical perspective, this task is equivalent to an optimal transportation problem. Given two discrete distributions $\mu = \{(x_i, \mu_i)\}_{i=1}^{m^{(s)}}$ and $\nu = \{(y_j, \nu_j)\}_{j=1}^{m^{(t)}}$, where x and y represent the positions of points, and μ and ν represent the mass weight of each point. Assuming that the total mass of μ and ν is equal. If the mass is effectively transferred from μ to ν, then the existence of a non negative entry matrix π is considered an effective transmission prototype. The OT problem is to find the optimal transportation matrix and minimize the total cost of transportation quality from a set of positions u to a set of positions ν based on the cost function $c(x_i, y_i)$:

$$D_u(\mu,\nu) = \inf_{\pi \in \prod(\mu,\nu)} \left\{ \sum_{ij} \pi_{ij} c(x_i, y_i) \right\} \tag{6}$$

where $\pi \in \prod(\mu, \nu)$ denotes the set of all viable transport prototypes.

Therefore, in the ALCDR method, the anchor link inference problem is solved by solving the OT problem, as shown in Figure 3. The user embeddings from two domains are considered as two discrete distributions in a shared vector space. By using the Sinkhorn method to solve the OT problem, the transition matrix T from the source domain to the target domain can be obtained.

$$\mathrm{T_{ts}} = \mathrm{SolveOT}(\mathrm{U_s}, \mathrm{U_T}) \in \mathbb{R}^{m^{(S)} \times m^{(t)}} \tag{7}$$

where $\mathrm{U_s}$ denotes the user embedding of source domain, $\mathrm{U_t}$ denotes the user embedding of target domain. Therefore, we can transport the cross-domain information according to the estimated OT plan by matrix multiplication:

$$\mathrm{U_{t \leftarrow s}} = \mathrm{T_{ts}} \mathrm{U_s} \in \mathbb{R}^{m^{(t)} \times d} \tag{8}$$

where the $\mathrm{U_{t\leftarrow s}}$ refers to the preserves information transport from the source domain to the target domain.

3.5 Inter-domain Aggregation

For user u in the target domain, three representations can be obtained: initialization embedding representation, graph convolutional embedding, and embedding representation obtained from inter class aggregation. A simple method is to directly connect the three embedding representations, but ALCDR uses a multi head self attention mechanism to fuse the three. The final embedding obtained from inter domain aggregation can be represented as:

$$\mathrm{M}_u^{(l)} = \mathrm{softmax}\left(\frac{\mathrm{Q}_u^{(l-1)} \mathrm{K}_u^{(l-1)\top}}{\sqrt{d}} \right) \mathrm{V}_u^{(l-1)} \tag{9}$$

$$\begin{cases} \mathrm{Q}_u^{(l-1)} = \mathrm{M}_u^{(l-1)} \mathrm{W}_{Q_u}^{(l-1)} \\ \mathrm{K}_u^{(l-1)} = \mathrm{M}_u^{(l-1)} \mathrm{W}_{K_u}^{(l-1)} \\ \mathrm{V}_u^{(l-1)} = \mathrm{M}_u^{(l-1)} \mathrm{W}_{V_u}^{(l-1)} \end{cases} \tag{10}$$

where $\mathrm{W}_{Q_u}^{(l-1)}, \mathrm{W}_{K_u}^{(l-1)}, \mathrm{W}_{V_u}^{(l-1)} \in \mathbb{R}^{d \times d}$ are learnable weight matrices.

Following previous works [11], we adopt the rating predictions as the optimization goal, which is formally defined as follows;

$$L_{BPR} = \sum_{u=1}^{m} \sum_{i \in N_u} \sum_{i \notin N_u} \ln \sigma(\hat{y}_{ij} - \hat{y}_{uj}) \tag{11}$$

Name	Parameter
Operating System	Linux3.10.0
CPU	Intel(R) Xeon(R) Gold 6226R
GPU	NVIDIA TESLA V100S
CUDA	11.0
python	3.6.5
pytorch	1.9.0
tensorflow	1.15.3

Table 1. Code implementation experimental environment

4 Implementation details

4.1 Code implementation

This reproduction is based on the method details explained in the author's paper. Since the author of the paper has not open sourced method project code and the dataset mentioned in the paper, I used the five public datasets in this experiment. The reproduction of this paper mainly completed the detailed code implementation of the ALCDR model proposed in the paper. At the same time, based on existing methods, it tried to use multiple domains as source domains of ALCDR to further improve cross-domain performance.

This reproduction relies on my own reading of the paper and understanding of the method proposed by the author, and I reproduce the code method on the premise of fully understanding its implementation process. Therefore, the code implementation is more difficult and there is more content that needs to be completed. The first thing to do is to implement its baseline effect in a single-domain model. The cross-domain method used in the paper is implemented based on LightGCN, so we also use this method as a single-domain baseline. Then we implement them one by one according to the three modules described in the paper. The first is Intra-domain Aggregation. Because this part uses the GNN method, we draw lessons from the implementation method of LightGCN. The second module is Anchor Link Inferring, this module is the core part of the ALCDR method. The main idea is to realize the transfer of knowledge by calculating the transfer matrix T from the source domain to the target domain. However, this module needs to be implemented by itself in the process of solving OT problem, that is, the SolveOT function. By referring to existing literature, the OT solution method was coded and implemented, and finally the approximate optimal solution of OT was obtained. Finally, in the Inter-domain Aggregation module, the author aggregates three different user representations through the self-attention mechanism, and I aggregate the learned user embeddings through code implementation. Finally, the overall framework of the model was realized. A lot of effort was also spent on code debugging and parameter adjustment. Fortunately, I finally fully reproduced the method and achieved effective performance improvements compared to the single-domain method on five public datasets.

4.2 Experimental environment setup

The detailed content of the experimental environment used for code reproduction is shown in the table 1. I conducted experiments on five public datasets, including Gowalla, Yelp2018, Yelp NC, Yelp-OH, and Amazon-

Dataset	User	Item	Interaction
Yelp-OH	5170	12997	185408
Yelp-NC	6336	13003	143884
Gowalla	29858	40981	1027370
Yelp2018	31668	38048	1561406
Amazon-Book	44709	46831	1174785

Table 2. Statistics of five datasets.

Book.

Gowalla: this is a sign in dataset obtained from the Gouwaila website [11] where users share locations through sign in. To ensure the quality of the dataset, we used 10-core settings [4] to retain users and items through at least 10 interactions.

Yelp-NC and **Yelp-OH** are records from North Carolina (NC) and Ohio (OH) in the Yelp2018 dataset, respectively, which contain a large number of user comments from local businesses.

Yelp2018: This dataset is adopted from the 2018 edition of the Yelp challenge. Wherein, the local businesses like restaurants and bars are viewed as the items. We use the same 10-core setting in order to ensure data quality.

Amazon-Book: Amazon reviews is a widely used product recommendation dataset. We choose Amazon books from our favorites. Similarly, we use 10-core settings to ensure that each user and item has at least 10 interactions.

We randomly select 80% of each user's historical interactions to form the training set, and use the remaining as the test set. From the training set, we randomly select 10% of interactions as the validation set to adjust hyperparameters. For each observed user item interaction, we consider it as a positive instance and then execute a negative sampling strategy to pair it with a negative item that the user has not previously consumed.

Table 2 exhibits the detailed statistics of the datasets. It is worth noting that although the number of overlapping users are existing in Yelp-OH and Yelp2018 and Yelp-NC and Yelp2018, ALCDR model does not use such information in the main results.

5 Results

In order to fully verify the effectiveness of code reproduction in the cross-domain experiment part, I used five public data sets and a total of 14 tasks for testing and verification.

The experimental group can be divided into five groups according to the source domain. Among the five data sets, Yelp-OH and Yelp-NC are small data sets compared to the other three data sets. 1)In the first set of tasks, Yelp-OH is used as the source domain, and the remaining four data sets are used as the target domain; 2)in the second set of tasks, Yelp-OH is used as the source domain, and the remaining four data sets are used as the target domain; 3)The third set of tasks Amazon-Book is used as the source domain, and the Yelp-OH and Yelp-NC data sets are used as the target domain; 4)the fourth group of tasks uses Yelp2018 as the source domain, and the Yelp-OH and Yelp-NC data sets are used as the target domain; 5)the fifth group of tasks Gowalla is used as the source domain, and the Yelp-OH and Yelp-NC data sets are used as the target domain.

Source domain	Target domain		LightGCN	ALCDR
Yelp-NC	Amazon-Book	Recall@20	0.07526	**0.07571**
		NDCG@20	0.03026	**0.03061**
	Yelp2018	Recall@20	0.06386	**0.06392**
		NDCG@20	0.05251	**0.05254**
	Gowalla	Recall@20	0.17950	**0.18012**
		NDCG@20	0.15309	**0.15380**
	Yelp-OH	Recall@20	0.07718	**0.07988**
		NDCG@20	0.02978	**0.03060**
Yelp-OH	Amazon-Book	Recall@20	0.07526	**0.07598**
		NDCG@20	0.03026	**0.03065**
	Yelp2018	Recall@20	0.06386	**0.06392**
		NDCG@20	0.05251	**0.05269**
	Gowalla	Recall@20	0.17950	**0.17990**
		NDCG@20	0.15309	**0.15355**
	Yelp-NC	Recall@20	0.06944	**0.07071**
		NDCG@20	0.02841	**0.02876**
Amazon-Book	Yelp-OH	Recall@20	0.07718	**0.07911**
		NDCG@20	0.02978	**0.03024**
	Yelp-NC	Recall@20	0.06944	**0.07039**
		NDCG@20	0.02841	**0.02890**
Yelp2018	Yelp-OH	Recall@20	0.07718	**0.07988**
		NDCG@20	0.02978	**0.03080**
	Yelp-NC	Recall@20	0.06944	**0.07086**
		NDCG@20	0.02841	**0.02920**
Gowalla	Yelp-OH	Recall@20	0.07718	**0.07969**
		NDCG@20	0.02978	**0.03050**
	Yelp-NC	Recall@20	0.06944	**0.07008**
		NDCG@20	0.02841	**0.02889**

Table 3. Recommendation performance of baseline approach. Best results are in bold. Improve denotes relative improvement over the best baseline.

Source domain	Target domain		LightGCN	ALCDR	ALCDR(**r**)
Yelp2018	Yelp-OH	Recall@20	0.07718	**0.07988**	0.07834
		NDCG@20	0.02978	**0.03080**	0.02978
	Yelp-NC	Recall@20	0.06944	**0.07086**	0.06818
		NDCG@20	0.02841	**0.02920**	0.02789
Gowalla	Yelp-OH	Recall@20	0.07718	**0.07969**	0.07795
		NDCG@20	0.02978	**0.03050**	0.02988
	Yelp-NC	Recall@20	0.06944	**0.07008**	0.06881
		NDCG@20	0.02841	**0.02889**	0.02797

Table 4. Recommendation performance of baseline approach. Best results are in bold. Improve denotes relative improvement over the best baseline. **(r)** refers to the random matrix T for ALCDR.

From the experimental results in the Table 3, we can easily see that compared with the recommendation performance of single-domain LightGCN, ALCDR has achieved effective improvements in five public data sets by introducing cross-domain knowledge. It also verified the effectiveness of this reproduction.

6 Analysis

Since the transfer matrix T proposed in the model is not intuitive, I conducted a comparative analysis experiment on T in this method. The transfer matrix T mentioned in ALCDR is calculated using source domain users and target domain users, and the matrix T is obtained by solving the OT problem. But there are some doubts about whether this matrix is really as effective as stated in the paper. Therefore, in this experiment, I randomly initialized a transfer matrix T to test its transfer effect. The experimental results are shown in Table 4.

I conducted experimental comparisons in the cross-domain recommendation tasks of four datasets. It can be seen from the experimental results that the effect of randomly initialized matrix T in four cross-domain tasks cannot improve the performance of the target domain like ALCDR. In some tasks, the performance is still weaker than single-domain LightGCN. Therefore, we can conclude that the transfer matrix T described in the paper does not simply improve performance by increasing model parameters. The use of source domain user embedding still plays a key role in the construction of matrix T, although it does not seem so intuitive.

7 Conclusion and future work

In this article, the author focuses on cross domain recommendation tasks to transfer knowledge from source domain to target domain. Most existing CDR methods assume that users have the same behavior pattern (interest expression) in different fields. In this article, the author believes that this assumption may not hold true in real life, as direct interest transfer between multiple domains for the same user may lead to noise in the learning process. At the same time, it was proposed to consider cross domain anchor chain connection as a learnable parameter, and a new ALCDR cross domain recommendation model based on OT transmission was ultimately designed, which accurately infers anchor chain paths from both intra domain and inter domain perspectives and deeply aggregates collaborative signals. And in this reproduction work, although the author did not disclose the source code and relevant information of the dataset, I conducted extensive experiments on four real-world public datasets to evaluate the ALCDR proposed in the paper, and the results verified its effectiveness.

References

[1] Hao Chen, Zhong Huang, Yue Xu, Zengde Deng, Feiran Huang, Peng He, and Zhoujun Li. Neighbor enhanced graph convolutional networks for node classification and recommendation. *Knowledge-Based Systems*, 246:108594, 2022.

[2] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[3] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. Herograph: A heterogeneous graph framework for multi-target cross-domain recommendation. In *ORSUM@ RecSys*, 2020.

[4] Ming He, Jiuling Zhang, Peng Yang, and Kaisheng Yao. Robust transfer learning for cross-domain collaborative filtering using multiple rating patterns approximation. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 225–233, 2018.

[5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

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

[7] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Twenty-First international joint conference on artificial intelligence*, 2009.

[8] Lile Li, Quan Do, and Wei Liu. Cross-domain recommendation via coupled factorization machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9965–9966, 2019.

[9] Pan Li and Alexander Tuzhilin. Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 331–339, 2020.

[10] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

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

[12] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[13] Yi Zhao, Chaozhuo Li, Jiquan Peng, Xiaohan Fang, Feiran Huang, Senzhang Wang, Xing Xie, and Jibing Gong. Beyond the overlapping users: cross-domain recommendation via adaptive anchor link learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1488–1497, 2023.

[14] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1507–1515, 2022.