Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 225 (2023) 2536–2545

Procedia Computer Science

www.elsevier.com/locate/procedia

27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2023)

meta-GRS: A Graph Neural Network for Cross-Domain Recommender System via Meta-Learning

Pham Minh Thu Do[a], Qian Zhang[a], Guangquan Zhang[a], Jie Lu[a,*]

[a]*Australian Artificial Intelligence Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, 2007, Australia*

Abstract

Despite the fact that Recommender Systems have been researched and developed for quite a while, they still hold some certain challenges, two of which are Cold Start and Data Sparsity. Being introduced as a savior, the Cross-domain Recommender system (CDR) owns the capacity to transfer knowledge across domains which makes it become the practical solution for the mentioned issues. In CDR approaches, the family of graph-based solutions is very effective, which builds graphs to illustrate the relationship between users, items, and other factors and learn their representations through graph representation learning. However, most graph-based approaches focus on extracting either domain-specific or domain-shared features and do not have the mechanism to prevent the transfer of private features, which can degrade the quality of vector representations. Moreover, the knowledge transfer process built on overlapping users makes the model biased toward these users, thus downgrading the performance on cold-start users. This paper proposes a meta-GRS framework that uses Graph Neural Network with Meta-Learning and Adversarial Learning for cross-domain recommendation. In meta-GRS, the representative user and item embeddings are improved thanks to the features extracted by the private and cross-domain graphs. Domain-shared features are learned under adversarial learning such that the domain discriminator is unable to determine whether the domain they came from can ensure the positive transfer. To optimize the model performance effectively, we use a Dynamic Weight Averaging algorithm to learn loss weights automatically. The model's parameters are optimized under the optimization-based meta-learning method provides our model the capability of generalization to the new users. Experiments on practical datasets illustrate that the meta-GRS leads the chart in the comparison of other state-of-the-art baselines in recommendation accuracy.

Keywords: recommendation; graph, neural network; meta-learning

* Corresponding author.
E-mail address: Jie.Lu@uts.edu.au

10.1016/j.procs.2023.10.245

1. INTRODUCTION

The past decade witnessed the explosive growth of the recommender system (RS) [1] has been applied in many fields. Collaborative Filtering is the dominating technique in building RS, assuming that people who share the same previous tastes will also have similar preferences later. However, single-domain RSs have two common issues: data sparsity and cold-start [2]. Introduced as a solution, a cross-domain recommender system (CDR) leverages information among domains to enhance performance and handle those long-standing problems. With new techniques, various ways are designed to transfer knowledge between domains. For example, EMCDR [3] learned the mapping function to establish the connection across domains, while DDTCDR [4] learned the orthogonal mapping function through dual metric learning. KerKT [5], which is the kernel-included method, ensured the transferred knowledge's consistency and handled the feature divergence. In current years, being inspired by the achievement of Graphs in exploiting user-item interactions, some researchers have tried using Graph to construct domain-specific relationships or establishing the connection between domains. GA-DTCDR [6] modeled within-domain relationships using a heterogeneous graph and used Node2vec [7] to embedded this graph while element-wise attention combined embedding vectors of overlapping users. HeroGraph [8] generated the domain-shared features by exploring the cross-domain graph using graph convolution operations and recurrent attention. PPGN [9] extracted common user features using graph convolution and propagation layers from the shared-domain graph. ReCDR [10] used Node2vec [7] to capture the private and shared characteristics from private graphs and shared-graph.

However, the family of graph-based solutions suffers from two problems. Firstly, most of the existing methods use graphs to extract either domain-specific features (GA-DTCDR [6]) or domain-shared features (HetoGraph [8], PPGN [9]). However, private and shared components are essential in describing vector representations. The ReCDR [10] method combined private and shared elements as vector representation, but it used Node2vec [7] to explore interaction graphs, which cannot capture the high-order user-item connectivity. While the ReCDR [10] extracted domain-specific and domain-shared features, they did not have the mechanism to prevent the transfer of private features. The domain-specific characteristics can be considered noises in the knowledge transfer [11, 12, 13], resulted in worse recommendation performance. Therefore, transferring domain-specific components is harmful and should be prevented. Secondly, these methods require significant overlapping user training data, making the model biased toward common users. As a consequence, the recommended items for cold-start users will be related to the common users, degrading the recommendation performance. Besides, current optimization-based meta-learning CDR frameworks [14, 15] follow the model-agnostic meta-learning (MAML) [16]. The MAML uses the manual learning rate, and the update direction follows the gradient.

To overcome these obstacles, we propose the cross-domain recommendation method called meta-GRS that uses Graph Neural Network with Meta-Learning and Adversarial Learning. To handle the first challenge, we first build within-domain graphs to depict user-item interactions in each domain. Then, based on overlapping users, the cross-domain graph is constructed to establish the connection between domains. We use a graph convolution network inspired by the lightGCN [17] to exploit those graphs. The features learned from the domain-specific graph are within-domain features, while domain-shared features are exploited from the cross-domain graph. Secondly, the adversarial learning includes the gradient reversal layer and domain classifier is used to prevent the transfer process of the domain-specific knowledge. To eliminate the second challenge, we use the meta-SGD method [18], an optimization-based meta-learning approach to optimize the model's parameters. In the local updating, the parameters are learned by optimizing the loss in the support set of each task, while the optimized parameters in global updating are learned through the optimization of losses in query sets of all tasks.

The main contributions are summarized:

- We improve the representative embedding vectors by extracting both domain-specific features and cross-domain features, in which domain-specific knowledge is the unique characteristic of a particular domain and domain-shared knowledge is meaningful for both domains. Moreover, we can enhance the quality of vector representations by preventing the domain-specific components from being transferred.
- We employ the meta-SGD [18] method to learn the model's parameters. Unlike the MAML method, meta-SGD has higher capability by learning the learner initialization, learning rate, and update direction. Moreover, we also

use the Dynamic Weight Averaging (DWA) algorithm [19] to learn the loss weights for each training phase. In this way, the model has a good generalization ability to adjust to cold-start users.

- Comprehensive experiments with meta-GRS and seven baselines on real-world datasets demonstrate the effectiveness of the proposed model.

The rest of this paper contains four sections. The following section reviews the relevant studies of single-domain RSs, cross-domain RSs, graph neural networks, and meta-learning. Section 3 defines the problem definition and preliminaries. Section 4 presents the structure of our meta-GRS in detail. Section 5 reports the experimental results between our model and seven baselines with the analysis. Section 6 includes the conclusion and several future directions.

2. RELATED WORKS

2.1. Single-domain recommendation

The explosive growth in information on the Internet provides an excellent chance to develop e-services applications in various domains, leading to difficulty retrieving the most relevant information to meet the customer's demands. Recommender Systems are information filtering tools that provide the most appropriate information to a particular user based on their needs [1, 2]. Single-domain RSs techniques can be categorized into two main approaches: Content-based (CB) and Collaborative Filtering (CF) [2]. A hybrid recommender system combines two techniques to leverage the benefits and reduce the drawbacks of each [20, 21].

In the CB RSs, the recommended items are close to the previous interest of users. Therefore, extracting features of items and user preference is the essential step. Recent CB RSs focus on using concept-based or semantic-aware representations. The semantics can be exploited from the unstructured data (endogenous method) or structured data (exogenous method). The endogenous methods concentrated on methods using embedding and distributed representations [22, 23]. On the other hand, recent developments in the exogenous method aim to capture the information encoded in the knowledge graph [24, 25].

Collaborative Filtering learns to make suggestions by exploiting the user's historical data with the vital point that the people who share previous preferences will also have the same tastes later. This category can be further clustered as memory-based or model-based approaches. The main task in the memory-based method is measuring user similarities (user-based) [1] or item similarities (item-based) [1] using rating data. On the other hand, the model-based approach uses data mining or machine learning to build the prediction model using historical data and other information. Deep Learning is also widely used in constructing RSs. It can be used to learn latent factors [26] or model the interactions [27].

2.2. Cross-domain recommendation

Single-domain recommendations have two long-standing issues affecting recommendation performance: cold-start and data sparsity [2]. The case in which no records of new objects is called the cold-start problem. Moreover, a minor number of historical records leads to the sparsity problem. As more and more users access multiple platforms makes the ability to build a system that leverages the data from different domains to enhance recommendation performance and handle the above issues. These systems are called Cross-domain Recommender Systems (CDRs) [2]. Compared with single-domain RSs, CDRs are more complicated since CDRs not only consider building the model to learn the user-item interactions within the domain but also have to transfer knowledge between domains. The knowledge between domains can be transmitted based on overlapping users. EMCDR [3] learned the mapping function to establish the connection across domains. The DDTCDR [4] learned the latent orthogonal mapping function through dual metric learning. The GA-DTCDR [6] designed a heterogenous graph to illustrate within-domain relationships and combine overlapping user representations using element-wise attention.

The domain-invariant features learned through the adversarial learning process are acted as a bridge or can be exchanged through domains. In the DA-CDR [11] and its updated version HMRec [12], an orthogonal constraint was applied to maximize the difference between the shared and within-domain subspaces. The gradient reversal layer and domain classifier ensured the latent feature spaces between both domains were maximally matched. The ALTRec [13] proposed the domain discriminator with adversarial learning to prevent the transferring of domain-specific features

2.3. Graph convolution networks in recommendation

With the inspiration of graph convolutional networks (GCN) [28], the user-item bipartite graphs are exploited using the propagation rule. In the single-domain RS, NGCF [29] followed the GCN rules to capture the high-order relatedness between items and users with various feature propagation layers. The LightGCN [17] observed the design of NGCF but removed the transformation function and the non-linear activation function since they have no positive impacts. Following the idea of using GCN in CDR, the PPGN [9] captured the user preferences through the multiple graph convolution and propagation layers exploited from the joint interaction graph. The HeroGraph [8] used a graph to model the cross-domain interactions while the private features were extracted from their IDs. ReCDR [10] constructed within-domain and cross-domain graphs, then applied Node2vec [7] to create node embeddings. They expanded new edges between nodes with higher similarity to enhance the information across domains. The BiTGCF [30] used graph collaborative filtering as the base model in their bi-direction transfer learning.

2.4. Meta-Learning

Meta-learning [31], has been applied in various research fields in recent years. It learns from many separate tasks (also known as metadata or meta-knowledge) to obtain a based model that can quickly generalize for unseen tasks. In the setting of meta-learning methods, there are meta-train task $\mathcal{D}_{train}$ and meta-test task $\mathcal{D}_{test}$. Each task is split into a training set called support set S_i, and an evaluation set named query set Q_i. While the learning objective at the level of each task is to boost the performance for unseen instances in the query set Q_i, this becomes getting a better performance for new jobs $\mathcal{D}_{test}$ at the level of multiple tasks. Specifically, the single task $\mathcal{T}_i$ has the support set $\mathcal{S}_i = (x_k, y_k)_{k=1}^{S}$ and the query set $Q_i = (x_k, y_k)_{k=1}^{Q}$, this task aims to learn a mapping function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$. The meta-learning approach uses meta-knowledge ω to ensure the effective learning of a single task $\mathcal{T}_i$ with the same mapping function f_θ and loss function $\mathcal{L}$. Meta-learning intends to optimize the ω, which can guide task-specific learning of all tasks to observe better performance. In TMCDR [14], meta-learning was used in the meta-stage to learn knowledge from common users and produced vector representations in the sparse domain for cold-start users. The meta-network was learned by optimizing the loss function across tasks. The PTUPCDR [32] designed the personalized bridge function for the transfer process. The customized mapping function between dense and sparse domains is generated with a meta-network. The CDAML [15] combined the meta-learning method and adversarial learning to alleviate the cold-start problems. The adversarial cross-domain approach was adopted into the meta-learning structure to extract and exchange domain-shared user preferences across domains.

3. PRELIMINARIES

3.1. Problem Definition

The symbol r_{ui} denotes the rating of user u to the item i and belongs to the rating matrix $\mathcal{R}$. The main task is to fulfill this matrix by predicting the missing ratings. We consider the cross-domain recommendation problem in which the more prosperous domain is the source domain while the sparse one is the target domain. Given the source domain $\mathcal{D}_\mathcal{S}$ and the target domain $\mathcal{D}_\mathcal{T}$, each domain has sets of users, items, and a rating matrix denoted as $\mathcal{S} = \{\mathcal{U}^\mathcal{S}, \mathcal{I}^\mathcal{S}, \mathcal{R}^\mathcal{S}\}$ and $\mathcal{T} = \{\mathcal{U}^\mathcal{T}, \mathcal{I}^\mathcal{T}, \mathcal{R}^\mathcal{T}\}$. The cross-domain recommendation task is to predict the unknown value of $\mathcal{R}^\mathcal{S}$, $\mathcal{R}^\mathcal{T}$. In our problem, we assume that two domains are shared the same users, so the $\mathcal{U}$ is a symbol for common users of two domains. Let n_u, $n_{i^\mathcal{S}}$, $n_{i^\mathcal{T}}$ denote the number of users, and items in the source and target domain, respectively.

3.2. Brief review of LightGCN

Graph convolution networks (GCN) [28] produce the feature representation of the target node by aggregating the features of its neighbors. LightGCN [17] is a GCN where the aggregation function is designed as follows:

$$e_u^{(k)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{(k-1)}$$
$$e_i^{(k)} = \sum_{i \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_u^{(k-1)} \tag{1}$$

Where $\mathcal{N}_u$ is the item set that user u interacted, and $\mathcal{N}_i$ is the user set that interacted with the item i, $e_u^{(k)}$ and $e_i^{(k)}$ are k-th layer embedding of user u and item i. The normalization term $\frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}$ can avoid the scale of embeddings increasing with the graph convolution operator.

4. PROPOSED FRAMEWORK

Figure 1 presents the proposed framework of meta-GRS. It is a dual-target cross-domain recommendation containing three components, taking the rating matrices as inputs and returning the predicted ratings as outputs. Three components are the domain-specific extractor, cross-domain extractor, and rating predictor.

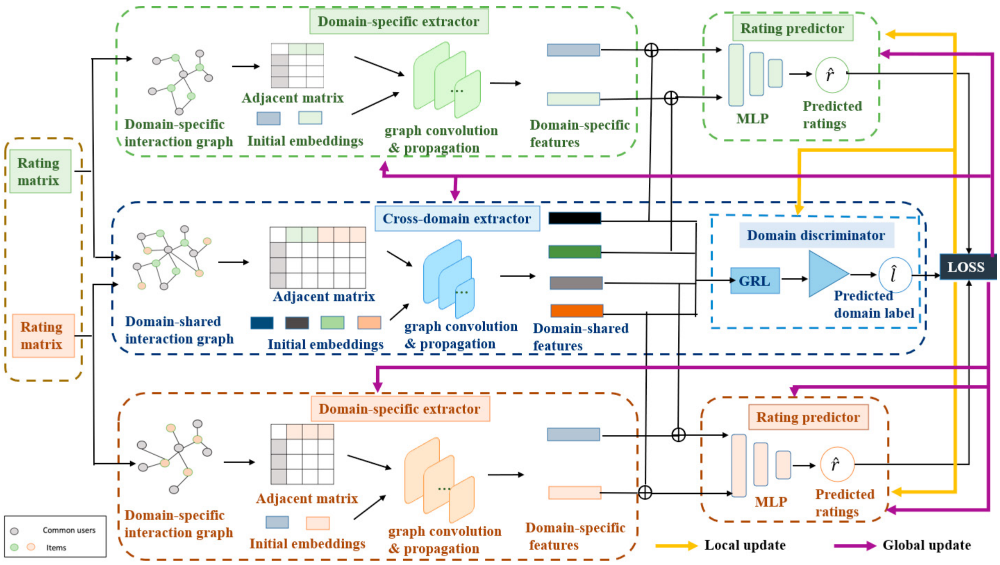


Fig. 1. The framework of the meta-GRS, a dual-target consisting of three components: 1) the domain-specific extractor, 2) cross-domain extractor and 3) rating predictor for each domain.

4.1. Domain-specific extractor

Firstly, we build a graph to represent the user-item interactions. We use the lightGCN layer to explore the high-order connections, which inputs the adjacent matrix and node embeddings. We construct the graph adjacent matrix where each element follows the rule:

$$\forall x \in X, x_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & otherwise \end{cases} \tag{2}$$

where $X \in \mathbb{R}^{(n_u+n_i)*(n_u+n_i)}$. Next, we use ID of users and items to create initial embedding vectors: $E_0 = [ep_u{}^{(0)}, ep_i{}^{(0)}] \in \mathbb{R}^{(n_u+n_i)*d_0}$, where d_0 is the initial embedding size. Through K times of propagation, we aggregate embeddings collected at each layer to produce the final embedding of a user/ an item. Then, we split it into two parts corresponding to the embedding vectors of items and users. These vectors are called domain-specific features since they describe the unique characteristics of each domain.

$$ep_u = \sum_{k=1}^{K} ep_u^{(k)}; ep_i = \sum_{k=1}^{K} ep_i^{(k)} \tag{3}$$

4.2. Cross-domain Extractor

We build the cross-domain graph to merge interactions in both domains. The adjacent matrix and initial embedding vectors $E_0 = [es_u{}^{(0)}, es_{i^S}{}^{(0)}, es_{i^T}{}^{(0)}]$ are put into the graph convolution layers, and the final representations are the combination of embeddings obtained at each layer. The adversarial learning with the gradient reversal layer and domain classifier, which takes the embedding vectors obtained from the propagation process as inputs and returns the domain labels, is used to prevent the transfer of domain-specific knowledge. We place the gradient reversal layer between the cross-domain extractor and the domain classifier. It can act as the identity function for forward propagation and reverse the gradient direction for backward propagation. Then the domain classifier uses the softmax function to predict the domain label.

4.3. Rating predictor

After extracting the domain-specific and cross-domain graphs, we can obtain private and shared features of users and items. The concatenation of private and shared features is the final presentation vector for users and items. Finally, we adopt the standard Multi-layer Perceptron to estimate the rating scores.

4.4. Optimization

Our objective function includes the mean square error ($\mathcal{L}_{p^S}, \mathcal{L}_{p^T}$) between the estimated rating scores and the observed rating score, and the cross-entropy loss $\mathcal{L}_c$ of the classification task during the adversarial learning.

$$\mathcal{L}_{p^S} = \sum_{u,i \in \mathcal{R}^S} (\hat{y}_{ui} - y_{ui})^2 \tag{4}$$

$$\mathcal{L}_{p^T} = \sum_{u,i \in \mathcal{R}^T} (\hat{y}_{ui} - y_{ui})^2 \tag{5}$$

$$\mathcal{L}_C = - \sum_{u=1}^{|\mathcal{U}^S|+|\mathcal{U}^T|} \hat{l}_u log l_u + (1-\hat{l}_u)\log(1-l_u) - \sum_{i=1}^{|\mathcal{I}^S|+|\mathcal{I}^T|} \hat{l}_i log l_i + (1-\hat{l}_i)\log(1-l_i) \tag{6}$$

where $\hat{y}_{ui}$ is the estimated rating score, $\hat{l}_u, \hat{l}_i$ are predicted domain labels of user u and item i, respectively. Finally, we have the objective function as follows:

$$\mathcal{L} = \alpha_1 \mathcal{L}_{p^S} + \alpha_2 \mathcal{L}_{p^T} + \alpha_3 \mathcal{L}_C + \lambda \|\Theta\|_2 \tag{7}$$

where Θ includes model parameters and embedding table for all users and items, the L2-regularization parameter parametrized by λ on Θ can prevent overfitting.

We employ the DWA algorithm to learn the weight of loss. The DWA algorithm [19] learns the weight based on the learning speed of each task with the rule that the lower the task's learning speed, the higher the task's loss weight. The learning speed of the task depends on the loss values over the last two training steps.

In the meta-learning setting, one task contains a user and the rated items of that user with corresponding rating scores in both domains. Each task has support sets and query sets.

Optimized parameters are produced using the idea of meta-SGD [18], which is known as an optimization-based meta-learning method. The update-needed parameters are categorized as follows:

$\theta^f = \{\theta^{f_{P_T}}, \theta^{f_{P_S}}, \theta^{f_S}\}$: are the user and item embedding weights in domain-specific and cross-domain extractor.

θ^D: parameters of the domain discriminator.

$\theta^{pre} = \{\theta^{pre_S}, \theta^{pre_T}\}$: parameters of the rating predictors.

As shown in Figure 1, the training process has two phases: local update and global update.

For the local update: in each training task, the support set is used in the local updating as follows:

$$\hat{\theta}_i^* \leftarrow \theta_i^* - \beta \circ \nabla_{\theta_i^*} \mathcal{L}_{\mathcal{T}_i}(., \theta_i^*) \tag{8}$$

Where $* = \{D, pre\}$, β is a vector that has the same size with θ^* that controls learning rate and update direction, and $\mathcal{L}_{\mathcal{T}_i}(., \theta_i^*)$ is the loss in the support set of task $\mathcal{T}_i$

For the global update: The sum of all losses in query sets of meta-training users are used in the global updating:

$$(\theta^*, \beta) \leftarrow (\theta^*, \beta) - \gamma \cdot \sum_{\mathcal{T}_i \in \mathcal{T}_{train}} \nabla_{(\theta_i^*, \beta)} \mathcal{L}'_{\mathcal{T}_i}(., \hat{\theta}_i^*) \tag{9}$$

Where $* = \{f, D, pre\}$, γ is the global updating learning rate, and $\mathcal{L}'_{\mathcal{T}_i}(., \hat{\theta}_i^*)$ is the loss in the query set of task T_i

In the evaluation, the base model is updated by a few interactions of new user. With the strong generalization ability of meta-learning and the advantage of knowledge transfer, the model can quickly adjust new users.

5. EXPERIMENTS

5.1. Datasets and Evaluation metrics

Datasets:The Amazon dataset (released in 2014) [1] is used for experiments. We select *CDs and Vinyl* (named music), *Movies and TV* (named movie), and *Books* (named book) for defining three recommendation tasks. We test our proposed framework in the scenarios of user overlapping. To simulate this scenario, we removed non-overlapped users. In the meta-learning setting, to simulate the cold-start scenario, We randomly sample 20% of users as cold-start users to form the meta-testing set and use the remaining as training data in the meta-training phase. We use 20% of the task data in each training task as the query set and the rest as the support set. For fairness, the support sets in the meta-testing phase are adopted as training data in the non-meta-learning method.

Table 1 shows the statistical information of each cross-domain task.

Table 1. Statistics of datasets

Task	Category	#users	#items	#ratings	Sparsity
Task 1	Music	2342	20857	115163	0.9976
	Movie	2342	31822	118544	0.9984
Task 2	Book	4426	123574	338318	0.9994
	Movie	4426	24055	176730	0.9983
Task 3	Book	2454	83468	172921	0.9992
	Music	2454	31358	106434	0.9986

Evaluation metrics: We employ Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as metrics.

5.2. Baselines and Settings

Baselines: We select the following benchmarks for comparison:

- **NeuMF** [27]: is the single domain recommendation that combined the hidden layers of Generalized Matrix Factorization and MLP to learn the interaction function.
- **NGCF** [29]: is the single-domain recommendation that used multiple feature propagation layers to capture the high-order relationship.
- **LightGCN** [17]: is the single-domain recommendation that observed the design NGCF but removed the transformation function and non-linear activation function.
- **EMCDR** [3]: is the cross-domain recommendation that learned the mapping function as the bridge between domains. It adopts Matrix Factorization (MF) to learn the embeddings.
- **PPGN** [9]: is the cross-domain recommendation that captured the user preferences from the shared graph using various graph convolution and propagation layers.
- **TMCDR** [14]: is the cross-domain recommendation that applied meta-learning on the EMCDR-based method.
- **PTUPCDR** [32] is the cross-domain recommendation that use meta-learning to build the personalized bridge function for the transfer process.

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

Experiment Settings: We implement our method using Pytorch. The source codes of NeuMF [2], NGCF[3], LightGCN[4], EMCDR[5], PPGN[6], and PTUPCDR[7] are open avaliable. We implement TMCDR base on the code of EMCDR. Some of them are evaluated using the ranking-based method (recall, precision, NDCG), so we rewrote to add the value-based metrics (RMSE, MAE). We use three hidden layers for our network structure's graph convolution and MLP parts. In our model, we run experiments ten times and take the average value.

5.3. Experimental Results

Comparison results of our proposed model with various benchmarks are shown in Table 2.

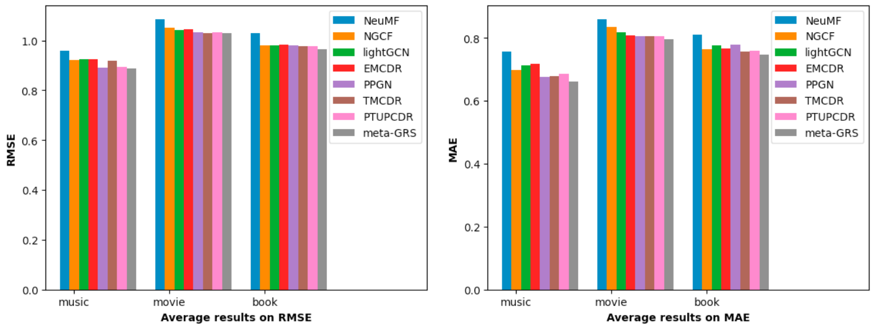
Table 2. We present the average results recorded throughout ten runs. The best outcomes are boldfaced, and the best baselines are underlined.

Task	**Category**	**Metrics**	**NeuMF**	**NGCF**	**LightGCN**	**EMCDR**	**PPGN**	**TMCDR**	**PTUPCDR**	**meta-GRS**
Task 1	Music	RMSE	0.9661	0.9202	0.9267	0.9311	0.8931	0.9225	0.8999	**0.8927**
		MAE	0.7610	0.6803	0.7041	0.7145	0.6720	0.6775	0.6859	**0.6710**
	Movie	RMSE	1.0937	1.0425	1.0342	1.0476	1.0306	1.0283	1.0313	**1.0246**
		MAE	0.8688	0.8210	0.8190	0.8108	0.7968	0.8076	0.7997	**0.7929**
Task 2	Book	RMSE	1.0494	0.9806	0.9821	0.9886	0.9799	0.9811	0.9841	**0.9758**
		MAE	0.8196	0.7695	0.7742	0.7731	**0.7604**	0.7627	0.7611	0.7618
	Movie	RMSE	1.0783	1.0595	1.0471	1.0441	1.0320	1.0337	**1.0318**	1.0347
		MAE	0.8504	0.8485	0.8151	0.8048	0.8130	0.8015	0.8126	**0.7977**
Task 3	Book	RMSE	1.0106	0.9805	0.9780	0.9808	0.9799	0.9743	0.9692	**0.9535**
		MAE	0.8024	0.7572	0.7759	0.7570	0.7939	0.7496	0.7588	**0.7335**
	Music	RMSE	0.9486	0.9252	0.9220	0.9213	0.8898	0.9153	0.8878	**0.8829**
		MAE	0.7539	0.7151	0.7182	0.7194	0.6777	0.6799	0.6871	**0.6495**

- **For the non-transfer recommendation methods**, the GCN-based methods (NGCF, LightGCN) outperforms NeuMF. The LightGCN behaves even better than the EMDR, a cross-domain recommendation method. The results validate that mining the high-order connectivity can return better embedding. However, the effectiveness of knowledge transfer has arisen in case data are getting sparser. Our meta-GRS records an improvement in RMSE of around 6% over NeuMF and about 3% over graph collaborative filtering methods.
- **For the cross-domain recommendation methods**, our meta-GRS performs better than other transfer methods in almost cases. The PPGN aims to extract the domain-shared features via a joint interaction graph and ignore the domain-specific features. At the same time, our method considers both private and shared features. The results prove that the domain-specific features are essential in describing the vector representations for users and items. The TMCDR method improves EMCDR in which the mapping function is learned under the optimization-based meta-learning method. Its better results prove the good generalization of the model learned via meta-learning. The high performance of PTUPCDR also shows that meta-learning is effective for cold-start recommendations. However, it is a single-target method, while our method can benefit both domains simultaneously.

We also compare the average results between our meta-GRS and all baselines in three categories. The results reported in Figure 2 show that meta-GRS has achieved the best performance in all categories.

[2] https://github.com/yihong-chen/neural-collaborative-filtering
[3] https://github.com/xiangwang1223/neural$_{g}raph_{c}ollaborative_{f}iltering$
[4] https://github.com/gusye1234/LightGCN-PyTorch
[5] https://github.com/MaJining92/EMCDR
[6] https://github.com/WHUIR/PPGN
[7] https://github.com/easezyc/WSDM2022-PTUPCDR

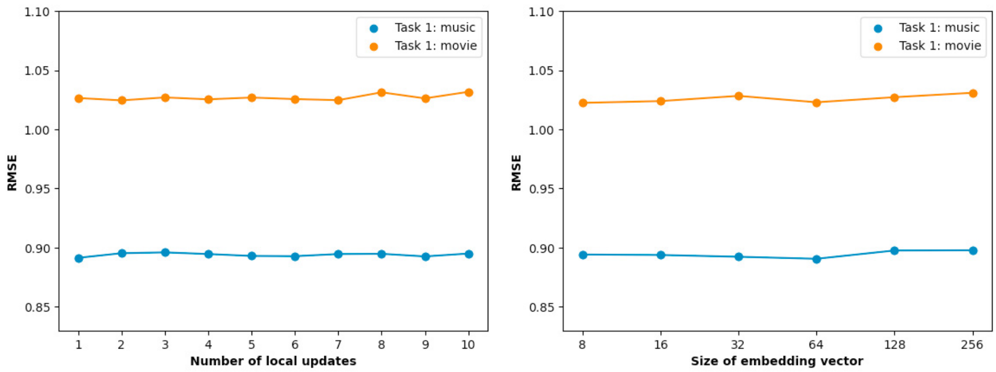


(a) (b)

Fig. 2. Average of (a) RMSE (b) MAE in three categories.

5.4. Parameters Analysis

We analyze the impact of parameters (the number of local updates, and the size of the embedding vector) as shown in Figure 3. As Figure 3(a) shows, performance is stable when the number of local updates increases. The results in Figure 3(b) illustrate that meta-GRS gains the best performance when the dimension is 64.



(a) (b)

Fig. 3. Parameter analysis of (a) number of local updates and (b) size of embedding vector in Task 1.

6. CONCLUSION

This paper proposes the graph neural network with meta-learning and dual adversarial network in CDR, named meta-GRS. It aims to extract useful domain-specific features while preventing them from being transferred. Using graph convolution layers, our model can capture the high-order connectivity of users and items. Combining the domain-specific and cross-domain knowledge returns valuable vector representations. With a good generalization ability, our meta-GRS can better adapt the cold-start users. Experiments show that meta-GRS can outperform state-of-the-art benchmarks in terms of RMSE and MAE. For future work, we extend our model to apply for multi-source recommendation and consider privacy-preserving to protect the sensitive information of each domain.

Acknowledgements

This work was supported by the Australian Research Council (ARC) under the Discovery Grant [DP220102635].

References

[1] Lu, Jie, Qian Zhang, and Guangquan Zhang. (2020) "Recommender Systems: Advanced Developments".

[2] Ricci, Francesco, Lior Rokach, and Bracha Shapira. (2022) "Recommender Systems: Techniques, Applications, and Challenges." 1-35.

[3] Man, Tong, et al. (2017) "Cross-domain recommendation: An embedding and mapping approach." *IJCAI* Vol. **17**.

[4] Li, Pan, and Alexander Tuzhilin. (2020) "Ddtcdr: Deep dual transfer cross domain recommendation." *Proceedings of the 13th International Conference on Web Search and Data Mining*.

[5] Zhang, Q., Lu, J., Wu, D., Zhang, G. (2018). "A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities". *IEEE transactions on neural networks and learning systems*, **30(7)**, 1998-2012.

[6] Zhu, Feng, et al. (2020) "A Graphical and Attentional Framework for Dual-Target Cross-Domain Recommendation." *IJCAI*.

[7] Grover, A., Leskovec, J. (2016, August). "node2vec: Scalable feature learning for networks". *In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*: 855-864.

[8] Cui, Qiang, et al. (2020) "HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation." *ORSUM@ RecSys*.

[9] Zhao, Cheng, Chenliang Li, and Cong Fu. (2019) "Cross-domain recommendation via preference propagation graphnet." *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

[10] Xu, Kun, et al. (2020) "Expanding Relationship for Cross Domain Recommendation." *Proceedings of the 30th ACM International Conference on Information Knowledge Management*.

[11] Zhang, Qian, et al. (2021) "A deep dual adversarial network for cross-domain recommendation." *IEEE Transactions on Knowledge and Data Engineering*.

[12] Liao, Wenhui, et al. (2022) "Heterogeneous Multidomain Recommender System Through Adversarial Learning." *IEEE Transactions on Neural Networks and Learning Systems*.

[13] Li, Ying, et al. (2020) "Atlrec: An attentional adversarial transfer learning network for cross-domain recommendation." *Journal of Computer Science and Technology* **35.4**: 794-808.

[14] Zhu, Yongchun, et al. (2021) "Transfer-meta framework for cross-domain recommendation to cold-start users." *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[15] Xu, Jiajie, et al. (2022) "CDAML: a cluster-based domain adaptive meta-learning model for cross domain recommendation." *World Wide Web*: 1-15.

[16] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. (2017) "Model-agnostic meta-learning for fast adaptation of deep networks." *International conference on machine learning*.

[17] He, Xiangnan, et al. (2020) "Lightgcn: Simplifying and powering graph convolution network for recommendation." *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*.

[18] Li, Z., Zhou, F., Chen, F., Li, H. (2017). "Meta-sgd: Learning to learn quickly for few-shot learning" *arXiv preprint arXiv:1707.09835*.

[19] Liu, S., Johns, E., Davison, A. J. (2019). "End-to-end multi-task learning with attention".*In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*: 1871-1880.

[20] Lu, J., Shambour, Q., Xu, Y., Lin, Q., Zhang, G. (2010). "BizSeeker: a hybrid semantic recommendation system for personalized government-to-business e-services". *Internet Research*.

[21] Lu, J., Shambour, Q., Xu, Y., Lin, Q., Zhang, G. (2013) " a web-based personalized business partner recommendation system using fuzzy semantic techniques". *Computational Intelligence*, **29(1)**, 37-69.

[22] Suglia, Alessandro, et al. (2017)"A deep architecture for content-based recommendations exploiting recurrent neural networks." *Proceedings of the 25th conference on user modeling, adaptation and personalization*.

[23] Musto, Cataldo, et al. (2018) "Deep content-based recommender systems exploiting recurrent neural networks and linked open data." *Adjunct Publication of the 26th conference on user modeling, adaptation and personalization*.

[24] Zhang, Fuzheng, et al. (2016) "Collaborative knowledge base embedding for recommender systems." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*.

[25] Wang, Hongwei, et al. (2019) "Knowledge graph convolutional networks for recommender systems." *The world wide web conference*.

[26] Lu, J., Xuan, J., Zhang, G., Luo, X. (2018). "Structural property-aware multilayer network embedding for latent factor analysis". *Pattern Recognition*, **76**, 228-241.

[27] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. S. (2017, April). "Neural collaborative filtering" . *In Proceedings of the 26th international conference on world wide web* (pp. 173-182).

[28] Zhang, Si, et al. (2019)"Graph convolutional networks: a comprehensive review." *Computational Social Networks* **6.1**: 1-23.

[29] Wang, Xiang, et al. (2019) "Neural graph collaborative filtering." *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*.

[30] Liu, Meng, et al. (2020) "Cross domain recommendation via bi-directional transfer graph collaborative filtering networks." *Proceedings of the 29th ACM International Conference on Information Knowledge Management*.

[31] Vilalta, Ricardo, and Youssef Drissi. (2002) "A perspective view and survey of meta-learning." *Artificial intelligence review* **18.2** : 77-95.

[32] Zhu, Yongchun, et al. (2022) "Personalized transfer of user preferences for cross-domain recommendation." *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*.