JSCN: Joint Spectral Convolutional Network for Cross Domain Recommendation

Zhiwei Liu[1], Lei Zheng[1], Jiawei Zhang[2], Jiayu Han[3], and Philip S. Yu[1]
[1]Department of Computer Science, University of Illinois at Chicago, IL, USA
{zliu213, lzheng21, psyu}@uic.edu
[2]IFM Lab, Department of Computer Science, Florida State University, FL, USA; jzhang@cs.fsu.edu
[3]Department of Computer Science, Jilin University, Changchun, China; jyhan15@mails.jlu.edu.cn

***Abstract*—Cross-domain recommendation can alleviate the data sparsity problem in recommender systems. To transfer the knowledge from one domain to another, one can either utilize the neighborhood information or learn a direct mapping function. However, all existing methods ignore the high-order connectivity information in cross-domain recommendation area and suffer from the domain-incompatibility problem. In this paper, we propose a Joint Spectral Convolutional Network (JSCN) for cross-domain recommendation. JSCN will simultaneously operate multi-layer spectral convolutions on different graphs, and jointly learn a domain-invariant user representation with a domain adaptive user mapping module. As a result, the high-order comprehensive connectivity information can be extracted by the spectral convolutions and the information can be transferred across domains with the domain-invariant user mapping. The domain adaptive user mapping module can help the incompatible domains to transfer the knowledge across each other. Extensive experiments on 24 Amazon rating datasets show the effectiveness of JSCN in the cross-domain recommendation, with 9.2% improvement on recall and 36.4% improvement on MAP compared with state-of-the-art methods. Our code is available online[1].**

***Index Terms*—Graph Convolutional Network, High-order Connectivity, Cross-domain Recommendation, Broad Learning**

I. Introduction

Recommending users with a set of preferred items is still an open problem [1]–[6], especially when the dataset is very sparse. To remedy the data sparsity issue, broad-leraning based model [7] and cross-domain recommender system [4], [8] are proposed where the information from other source domains can be transferred to the target domain. To transfer the knowledge from one domain to another, one can use the overlapping users [4], [6], [8], [9] in two ways: (1) the neighborhood information of common users stores the structure information of different domains with which we can do cross-domain recommendation [6], [10]; or (2) we can learn a mapping function [4], [8] to project latent vectors learned in one domain into another, and thus the knowledge can be transferred.

However, all existing methods ignore the *high-order connectivity* information [11]. High-order connectivity information consists of all the neighborhood information, the neighbors of all the neighbors, and so on by using the linkage information in the graph. The high-order connectivity information is explained in Figure 1 wherein the middle part user A and user C are the overlapping users, the upper/green part is the target domain, and the lower/blue part is the source domain. For example, in the target domain *(only the upper part)*, user D has a connection with item 4. Merely with the neighbor-based information [3], [8], [12], item 1 and item 2 should be ranked similarly since the neighbor user (i.e. user C) of user D has no direct connections with them. However, with the high-order connectivity information, we argue that user D should prefer item 2 more than item 1 as there is a path from item 2 to user D , while item 1 is only connected with user A and apart from the others. Moreover, the preference ranking may be different if taking account of the source domain *(considering both the upper and lower graphs)*. We can find two paths from item 1 to user D compared with the single path from item 2 to user D. Hence user D may prefer item 1 more than item 2 if the high-order connectivity information across domains is included. However, the high-order connectivity problem is not well studied yet in the cross-domain recommendation.

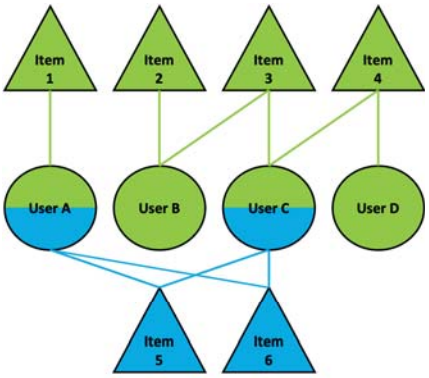


Fig. 1: A toy example of high-order connectivity information in cross-domain recommendation. The upper/green part is the target domain and the below/blue part is the source domain.

To capture the connectivity information in a graph, one can transform the graph into the frequency domain by applying the spectral graph theory [11], [13], [14]. In spectral theory [11], [15], the spectrum of a graph extracts the comprehensive connectivity information of a graph with the graph Fourier transformer in terms of the eigenvectors of

[1]https://github.com/JimLiu96/JSCN

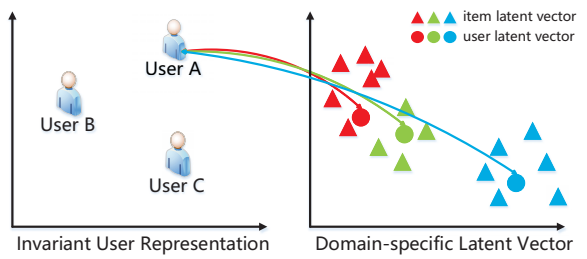


Fig. 2: Mapping users to different domains. Each user has domain invariant user representation on the left, which is projected as different domain-specific latent vectors over a specific domain on the right. Different colors on the right represent different domains.

the graph laplacian [11]. Based on this, we can design the spectral convolutional network [2], [16] whose convolutions are linear operators diagonalized by the Fourier basis. With the spectral convolutional network, nodes in a graph are represented as spectral vectors [2], [13]. When it comes to bipartite graphs, we can learn the spectral representations of users and items to capture the connectivity information. The spectral representation models the high-order non-linear interactions among users and items with multi-layer spectral convolutions. Hence, recalling the problem discussed before in Figure 1, in the spectral domain, the item 1 will be closer to user D than item 2 as there exist more connections from item 1 to user D than those from item 2.

However, different domains may be incompatible with each other which is also called as *domain-incompatibility* problem [17] in the cross-domain recommendation. For instance, if the target domain is a *Movie* domain where users are connected with the movie items, and the source domain is a *Clothing* domain where users are connected with the clothing items, they will be incompatible with each other since the behavior of users varies a lot. The information from the source domain cannot be directly utilized in the target domain. Thus we need to propose some mapping methods [4], [18], [19] as a bridge for the information transferring.

In this paper, unlike previous direct mapping methods [4], [10], [18], we view the latent vectors of a user in a specific domain as an interest projection from a *domain-invariant representation*. We show an illustration of mapping the domain-invariant user representation to a domain-specific user latent vectors in Fig. 2. To learn transferable representations, we jointly learn the domain-invariant representation of users across different domains. The joint convolution can capture the high-order connectivity information across different domains and learn domain-invariant representations by keeping the spectral similarity of the overlapping users. Based on this, we design a Joint Spectral Convolutional Network (JSCN) to fuse the information from multiple domains. JSCN will simultaneously operate multi-layer spectral convolution on the graph from each domain. Then the extracted spectral features can be shared across different graphs with the domain-invariant representations. Since JSCN jointly learns the spectral representations on different graphs, the high-order comprehensive connectivity information can be shared across domains. And because of the domain-invariant representations of users, JSCN alleviates the domain-incompatibility problem. We summarize our main contributions as follows:

- **Transferable spectral representation:** To the best of our knowledge, it is the first work to study how to transfer the spectral representation of bipartite graphs, which captures the high-order non-linear interactions of user-item both within domain and across domains.
- **Joint spectral convolution on graphs:** In this paper, we design a joint spectral convolutional network for learning the representations of multiple graphs concurrently. The high-order comprehensive connectivity information can be shared across different graphs.
- **Domain adaptive module:** To deal with the domain incompatibility problem, we apply a novel domain adaptive module to jointly learn the domain-invariant spectral representations of users, with which we can implement the joint convolution on graphs and share information across different domains.

The rest of the paper is organized as follows. In Sec. II, we review some previous works related to this paper. Then in Sec. III, we introduce the definitions of the notations and concepts, as well as the problem. In Sec. IV, we present the proposed model and the formulation of the model. Finally, in Sec. V we discuss the experiment before we draw a conclusion in Sec. VI.

II. Related Work

In this section we give a brief review of two closely related areas: (1) deep learning based recommender system; and (2) cross-domain Recommendation.

A. Deep learning based recommender system

Since [20] introduces deep learning into recommender system (RS), [3], [21], [22] propose deep neural network based RS to learn from either explicit or implicit data. To counter the sparsity problem, some scholars propose to utilize deep learning techniques to build a hybrid recommender system. [23] and [24] introduce Convolutional Neural Networks (CNN) and Deep Belief Network (DBN) assist in representation learning for music data. These approaches above pre-train embeddings of users and items with matrix factorization and utilize deep models to fine-tune the learned item features based on item content. In [5], a multi-view deep model is built to utilize item information from more than one domain. [25] integrates a CNN with PMF to analyze documents associated with items to predict users' future explicit ratings. [26] leverages two parallel neural networks to jointly model latent factors of users and items. To incorporate visual signals into RS, [27]–[30] propose CNN-based models to incorporate visual signals into RS. They make use of visual features extracted from product images using deep networks to enhance the performance of RS. [12], [31] investigates how to leverage the multi-view information to improve the quality of recommender systems.

Due to the limited space, readers can refer to [32] for more works on deep recommender systems.

B. Cross-domain recommendation and broad learning

Broad Learning [7] is a way to transfer the information from different domains, which focuses on fusing and mining multiple information sources of large volumes and diverse varieties. To solve the *cold-start* problem in item recommendation, cross-domain recommendation is proposed by either learning shallow embedding with factorization machine [8], [10], [33], [34] or learning deep embedding with neural networks [4], [9], [35]–[37]. When learning shallow embedding, CMF [33] jointly factorizes the user-item interaction matrices from different domains. In order to model the domain information explicitly, CDTF [8] and CDCF [34] is designed where the former factorizes the *user-item-domain* triadic relation and the later models the source domain information as the context information of users. When learning the deep embedding of users and items, CSN [35] is introduced firstly in multi-task learning scenario, where a convolutional network with cross-stitch units can share the parameters across different domains. This idea is extended later by CoNet [9] with cross connections across different networks where shared mapping matrices is introduced to transfer the knowledge. Additionally, EMCDR [4] transfers the knowledge across source and target domains with multi-layer perceptron. Our proposed JSCN model also jointly learns a deep embedding for both in-domain and cross-domain information.

III. Preliminaries and Definition

In this section, the preliminaries and definitions are presented. At first, we formally define the user-item bipartite graph and the corresponding connectivity matrices. Then we define the bipartite graph domain as well as the source domain and target domain before we formulate our problem. The important notations used in this paper are summarized in Table I.

TABLE I: Important notations

Notation	Description
$\mathcal{B}, \mathcal{B}^s, \mathcal{B}^t$	bipartite graph, source graph, target graph
$\mathcal{U}$	set of users
$\mathcal{I}$	set of items
u, i	user, item
$\widetilde{\mathcal{U}}$	set of common users
$\widetilde{u}$	common user
$\mho, \Lambda$	eigenvectors, diag-matrix of eigenvalues
C	input dimension of feature vector
F	spectral convolution parameter in each layer
$\mathbf{V}^u$,$\mathbf{V}^i$	user, item latent vectors
$\mathbf{U}^s$,$\mathbf{U}^t$	source, target invariant user representation
d	dimension of spectral latent vectors
d'	dimension of domain-invariant representation
$\Phi_{\mathcal{B}}$	domain related user mapping function
Ψ	categorical mapping function of items

Definition 1: **(Bipartite Graph)**. A bipartite user-item graph $\mathcal{B}$ with N vertices and E edges for recommendation is defined as $\mathcal{B} = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$, where $\mathcal{U}$ and $\mathcal{I}$ are two disjoint vertex sets, i.e. user set and item set, respectively. Every edge $e \in \mathcal{E}$ is in the form as $e = (u, i)$, denoting the interaction of a user $u \in \mathcal{U}$ with an item $i \in \mathcal{I}$, e.g. an item is viewed/purchased/liked by a user.

A bipartite graph describes the interactions among users and items, thus we can define an *implicit feedback matrix* [3], [38] $\mathbf{R} \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{I}|}$ for a given bipartite graph $\mathcal{B}$ as

$$\mathbf{R}_{r,j} = \begin{cases} 1 & \text{if } (u_r, i_j) \text{ interaction is observed} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where u_r and i_j are the r-th user in the user set $\mathcal{U}$ and j-th item in the item set $\mathcal{I}$, respectively.

Given an implicit feedback matrix $\mathbf{R}$ of a bipartite graph $\mathcal{B}$, the corresponding *adjacent matrix* $\mathbf{A}$ can be defined as

$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{R} \\ \mathbf{R}^\top & 0 \end{bmatrix}, \tag{2}$$

where the adjacent matrix $\mathbf{A}$ is an $N \times N$ matrix and N is the number of nodes in the bipartite graph, i.e., $N = |\mathcal{U}| + |\mathcal{I}|$.

With the adjacent matrix of a bipartite graph, a *laplacian matrix* $\mathbf{L}$ of a bipartite graph can be calculated as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}, \tag{3}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{D}$ is a diagonal matrix where each entry on the diagonal denotes the sum of all the elements in the corresponding row of the adjacent matrix, i.e. $\mathbf{D}_{k,k} = \sum_t \mathbf{A}_{k,t}$.

In this paper, we focus on the cross-domain recommendation. Thus we would combine the information from a set of bipartite graphs and then recommend items to users. In each domain, we have a categorical mapping function Ψ which projects the items into a specific category, e.g. *Movies*, describing the type of the items in the domain. We assume all the items belongs to one domain and thus we have the definition of graph domain.

Definition 2: **(Bipartite graph domain)** A Bipartite graph domain is defined on a categorical mapping function Ψ of items. Two bipartite graphs $\mathcal{B}_1$ and $\mathcal{B}_2$ are in different domains if and only if $\Psi(\mathcal{I}_1) \neq \Psi(\mathcal{I}_2)$.

The **source domain** bipartite graph is the source interaction bipartite graph of users and items, which provides auxiliary information for **target domain** bipartite graph where we would recommend items to users. We would integrate the information across the source domain and target domain, and make a recommendation in the target domain.

Definition 3: **(Problem Definition)**. Given a set of source domain bipartite graphs $\{\mathcal{B}_1^s, \mathcal{B}_2^s, ..., \mathcal{B}_M^s\}$ and a target domain graph $\mathcal{B}^t$, we aim at recommending each user in $\mathcal{U}^t$ with a ranked list of items from $\mathcal{I}^t$ which have no existing interaction with that user in graph $\mathcal{B}^t$. The source domains share a set of common users with each other, and the shared users between pairwise source domains can be denoted as set $\{\widetilde{\mathcal{U}}_{12}^s, \widetilde{\mathcal{U}}_{13}^s, ..., \widetilde{\mathcal{U}}_{(M-1)M}^s\}$. Meanwhile, the target domain also shares a set of common users with each of the source domains, which denoted as set $\{\widetilde{\mathcal{U}}_1^t, \widetilde{\mathcal{U}}_2^t, ..., \widetilde{\mathcal{U}}_M^t\}$.

IV. Proposed Model

In this section, we explain the spectral convolution network for collaborative filtering [2] first before we introduce the do-

main invariant user representation. After that, we will present our proposed **J**oint **S**pectral **C**onvolution **N**etwork (JSCN) for cross-domain recommendation. Finally, we will formulate the adaptive user mapping mechanism. The overall framework of our proposed model is given in Fig. 3. We use triangles and squares denoting users and items, respectively. Different colors for users and items denote different domains. And the same numbers on squares represent common users in different domains.

A. Spectral Convolution on Graph

Given a bipartite graph $\mathcal{B} = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$, we would like to learn an embedding for each of the node, i.e. user or item, as illustrated in the first step in Fig. 3. At first, users and items are represented as C-dimensional vectors, and all the user and item latent vectors can be grouped together and represented as matrices $\mathbf{X}^u$ and $\mathbf{X}^i$ respectively, where $\mathbf{X}^u \in \mathbb{R}^{|\mathcal{U}|\times C}$ and $\mathbf{X}^i \in \mathbb{R}^{|\mathcal{I}|\times C}$. With the graph structure information, the spectral convolutional operator $sp(\cdot)$ is defined [2], [11], [14] based on the eigenvectors $\mho = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ and the corresonding eigenvalues $\Lambda = diag\{\lambda_0, \lambda_1, \ldots, \lambda_{N-1}\}$ as

$$\begin{bmatrix}\mathbf{X}^{u*}\\ \mathbf{X}^{i*}\end{bmatrix} = sp(\begin{bmatrix}\mathbf{X}^{u}\\ \mathbf{X}^{i}\end{bmatrix}; \mho, \Lambda, \Theta) = \sigma\left((\mho\mho^\top + \mho\Lambda\mho^\top)\begin{bmatrix}\mathbf{X}^{u}\\ \mathbf{X}^{i}\end{bmatrix}\Theta\right) \tag{4}$$

In Eq. (4), the $\mho\mho^\top + \mho\Lambda\mho^\top$ term preserves the structure information of the bipartite graph, where $\Theta \in \mathbb{R}^{C\times F}$ is the convolutional filter to extract the spectral feature, and $\sigma(\cdot)$ denotes the logistic sigmoid function. It is the SP layer in the second step in Fig. 3.

With multiple spectral convolutional operators on the original feature vectors $[\mathbf{X}_0^u, \mathbf{X}_0^i]^\top$, we construct a K-layer spectral convolutional network on bipartite graph as shown in Eq. (5), with which we could learn the spectral representations of the nodes in the graph,

$$\begin{bmatrix}\mathbf{X}_K^{u}\\ \mathbf{X}_K^{i}\end{bmatrix} = \underbrace{sp(\ldots sp(}_{K}\begin{bmatrix}\mathbf{X}_0^{u}\\ \mathbf{X}_0^{i}\end{bmatrix}; \mho, \Lambda, \Theta_0)\ldots\mho, \Lambda, \Theta_{K-1}), \tag{5}$$

where $\Theta_0 \in \mathbb{R}^{C\times F}$ and $\Theta_k \in \mathbb{R}^{F\times F}$ $(k = 1, 2, \ldots, K-1)$. After K-layer spectral convolutional operations, we represent the users and items as latent vectors $\mathbf{V}^u \in \mathbb{R}^{|\mathcal{U}|\times d}$ and $\mathbf{V}^i \in \mathbb{R}^{|\mathcal{I}|\times d}$ respectively by either concatenating the extracted spectral features vectors at each layer or using the spectral feature vectors at the last layer. It corresponds to the third step in Fig. 3.

In terms of the loss function, we apply the *BPR*-loss as suggested in [2], [38] to compute the *in-domain* loss, which models the in-domain user-item interactions,

$$\mathcal{L}_i = \sum_{(r,j,j')\in\mathcal{D}} -\ln\sigma\left(\mathbf{V}^u(u_r)\cdot\mathbf{V}^i(i_j) - \mathbf{V}^u(u_r)\cdot\mathbf{V}^i(i_{j'})\right). \tag{6}$$

where the $\{(r, j, j')\}$ are the triples that sampled from user-item interaction records $\mathcal{D}$ in which r denotes the index of a user, j denotes the index of an item with which the user has interaction, and j' denotes the index of an item with which the user has no interaction. And we apply dot product $\cdot$ of user vector and item vector. Unlike pair-wise learning process [39], *BPR*-loss maximizes the difference between (r, j) and (r, j') with the assumption that users prefer observed items i_j over unobserved items $i_{j'}$. We use $\mathbf{V}^u(u_r)$ denotes the user latent vector of user u_r, and $\mathbf{V}^i(i_j)$ and $\mathbf{V}^i(i_{j'})$ denote the item latent vector of item i_j and item $i_{j'}$, respectively.

B. Domain Invariant User Representation

With the in-domain loss $\mathcal{L}_i$, we could learn both the user and item latent vectors from the multi-layer spectral convolutional network. Recall the problem definition in Def. 3, we have a set of source domain bipartite graphs $\{\mathcal{B}_1^s, \mathcal{B}_2^s, ..., \mathcal{B}_M^s\}$ and one target domain bipartite graph $\mathcal{B}^t$, and every domain has a set of overlapping users with each other.

A user requires different aspects w.r.t. different domains that lead to different user latent vectors, but we prefer invariant user representation across different domains, and hence we define the domain invariant user representation as $\mathbf{U} \in \mathbb{R}^{|\mathcal{U}|\times d'}$, from which we generate the domain-specific latent vector $\mathbf{V}^u$ with corresponding domain-related user mapping function $\Phi_{\mathcal{B}}$ as $\mathbf{V}^u = \Phi_{\mathcal{B}}(\mathbf{U})$.

For example, $\mathcal{B}_m^s = \{\mathcal{U}_m^s, \mathcal{I}_m^s, \mathcal{E}_m^s\}$ has a set of common users with the target domain $\mathcal{B}^t = \{\mathcal{U}^t, \mathcal{I}^t, \mathcal{E}^t\}$, which is denoted as $\widetilde{\mathcal{U}}_m^t$. With the in-domain loss, we learn the domain specific user latent vectors individually for $\mathcal{B}_m^s$ and $\mathcal{B}^t$ as $\mathbf{V}_m^{us}$ and $\mathbf{V}^{ut}$ respectively. $\mathbf{V}_m^{us}$ is generated from the domain-independent user representations $\mathbf{U}_m^s$ by the corresponding domain-related user mapping function $\Phi_{\mathcal{B}_m^s}$. $\mathbf{V}^{ut}$ is generated from the domain-independent user representations $\mathbf{U}^t$ by the corresponding domain-related user mapping function $\Phi_{\mathcal{B}^t}$. With the inverse function of the user mapping function Φ, denoted as Φ^{-1}, we can obtain the domain invariant user representation from the domain specific user latent vector as $\mathbf{U} = \Phi_{\mathcal{B}}^{-1}(\mathbf{V}^u)$, which is the fourth step in Fig. 3.

Since we have the domain invariant user representations, each user in $\widetilde{\mathcal{U}}_m^t$ should be represented as a same representation both in $\mathbf{U}_m^s$ and $\mathbf{U}^t$. To make this constraint trainable, we construct the *cross-domain* loss $\mathcal{L}_c$ as the l_2 distance of the domain invariant user representations as:

$$\begin{aligned}\mathcal{L}_c = &\sum_{m=1}^{(M-1)}\sum_{n=m+1}^{M}\sum_{\widetilde{u}\in\widetilde{\mathcal{U}}_{mn}^s}\|\mathbf{U}_m^s(\widetilde{u}) - \mathbf{U}_n^s(\widetilde{u})\|_2\\ &+\sum_{m=1}^{M}\sum_{\widetilde{u}\in\widetilde{\mathcal{U}}_m^t}\left\|\mathbf{U}_m^s(\widetilde{u}) - \mathbf{U}^t(\widetilde{u})\right\|_2,\end{aligned} \tag{7}$$

where $\widetilde{u} \in \widetilde{\mathcal{U}}_{mn}^s$ denotes the common users between source domains m and n as defined in Def. (3). $\mathbf{U}_m^s(\widetilde{u})$, $\mathbf{U}_n^s(\widetilde{u})$ and $\mathbf{U}^t(\widetilde{u})$ denotes the domain invariant representation of the anchor user $\widetilde{u}$ w.r.t. the corresponding domain-independent user representations $\mathbf{U}_m^s$, $\mathbf{U}_n^s$ and $\mathbf{U}^t$, respectively.

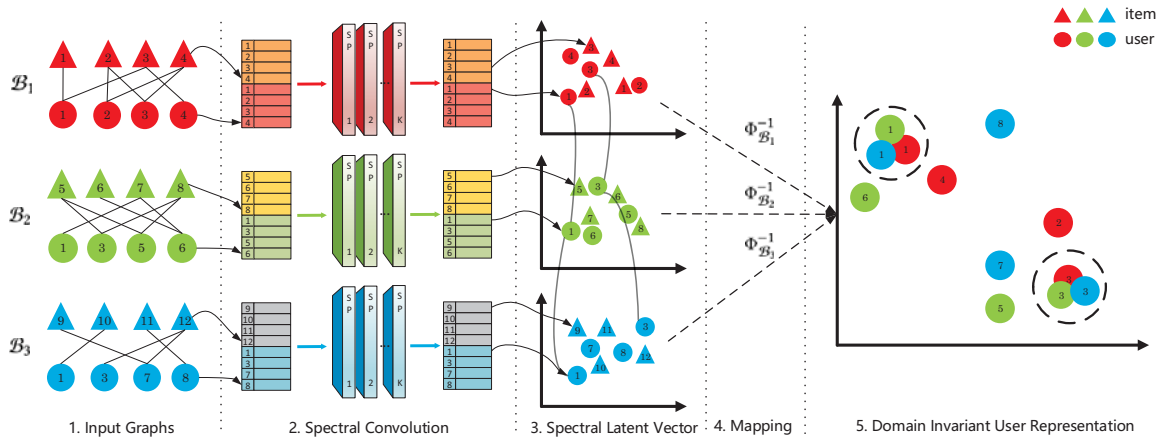


Fig. 3: The framework of training joint spectral convolution network (JSCN) model. At first, we randomly initialize the users (below part) and items (upper part) in the input graphs. Secondly, we learn spectral latent vectors of users and item with K-layer spectral convolution network (SP). Then we map the spectral latent vectors of users to domain invariant user representations with user mapping function $\Phi_{\mathcal{B}_i}^{-1}$. Finally, we minimize the distance of common users in domain invariant user representation space.

C. Joint Spectral Convolutional Network

The cross-domain loss $\mathcal{L}_c$ combines the information across different domains with the domain invariant user representation of the common users. Even if a common user only exists in part of all the domains, the information can be shared across different domains, as the effect of collaborative filtering. But we cannot directly learn the domain invariant representation, and thus instead, we learn the user and item latent vectors with the in-domain loss. Then we apply the inverse function of the user mapping function to learn the domain-invariant user representations. And the cross-domain loss can be written as:

$$\mathcal{L}_c = \sum_{m=1}^{(M-1)} \sum_{n=m+1}^{M} \sum_{\widetilde{u} \in \widetilde{\mathcal{U}}_{mn}^s} \left\| \Phi_{\mathcal{B}_m^s}^{-1}\left(\mathbf{V}_m^{us}(\widetilde{u})\right) - \Phi_{\mathcal{B}_n^s}^{-1}\left(\mathbf{V}_n^{us}(\widetilde{u})\right) \right\|_2^2 + \sum_{m=1}^{M} \sum_{\widetilde{u} \in \widetilde{\mathcal{U}}_m^t} \left\| \Phi_{\mathcal{B}_m^s}^{-1}\left(\mathbf{V}_m^{us}(\widetilde{u})\right) - \Phi_{\mathcal{B}^t}^{-1}\left(\mathbf{V}^{t}(\widetilde{u})\right) \right\|_2^2, \tag{8}$$

where $\mathbf{V}_m^{us}(\widetilde{u})$, $\mathbf{V}_n^{us}(\widetilde{u})$ and $\mathbf{V}^{ut}(\widetilde{u})$ denote the latent vector of the common user $\widetilde{u}$ w.r.t. the corresponding domain-specific user latent vectors $\mathbf{V}_m^s$, $\mathbf{V}_n^s$ and $\mathbf{V}^t$, respectively. We present this in the fifth step in Fig. 3. Hence the joint spectral convolution model has the loss function as:

$$\mathcal{L} = \sum_{m=1}^{M} \mathcal{L}_{im}^s + \mathcal{L}_i^t + \mathcal{L}_c + Reg, \tag{9}$$

where $\mathcal{L}_{im}^s$ is the in-domain loss of the source domain B_m^s, $\mathcal{L}_i^t$ is the in-domain loss of the target domain B^t, and Reg is the regularization term defined as:

$$Reg = \epsilon \left(\sum_{m=1}^{M} \|\mathbf{V}_m^{us}\|_2^2 + \|\mathbf{V}^t\|_2^2 \right), \tag{10}$$

where ϵ is the regularization hyper-parameter.

D. Adaptive User Mapping Module

As described in Sec. IV-B, we can use the inverse function of the domain-related user mapping function to generate the domain-invariant user representation from the spectral user latent vector. We define this inverse function as the adaptive user mapping function, which can either be a linear mapping function or a neural network based non-linear function [3]. For simplicity, here we only present the linear mapping function, which leads to

$$\mathbf{U} = \mathbf{V}^u W_{\mathcal{B}}, \tag{11}$$

where the $W_{\mathcal{B}}$ is the domain adaptive matrix w.r.t. graph domain $\mathcal{B}$. This mapping function is a kind of structural regularization [40] of different domains. It turns out the mapping can transfer the spectral information during the joint learning process.

With this adaptive user mapping matrix, we can rewrite the cross-domain loss as:

$$\mathcal{L}_c = \sum_{m=1}^{(M-1)} \sum_{n=m+1}^{M} \sum_{\widetilde{u} \in \widetilde{\mathcal{U}}_{mn}^s} \left\| \mathbf{V}_m^{us}(\widetilde{u}) W_{\mathcal{B}_m^s} - \mathbf{V}_n^{us}(\widetilde{u}) W_{\mathcal{B}_n^s} \right\|_2^2 + \sum_{m=1}^{M} \sum_{\widetilde{u} \in \widetilde{\mathcal{U}}_m^t} \left\| \mathbf{V}_m^{us}(\widetilde{u}) W_{\mathcal{B}_m^s} - \mathbf{V}^{t}(\widetilde{u}) W_{\mathcal{B}^t} \right\|_2^2, \tag{12}$$

where $W_{\mathcal{B}_m^s}$ and $W_{\mathcal{B}_n^s}$ are two adaptive user mapping matrix corresponding with the domain $\mathcal{B}_m^s$ and $\mathcal{B}_n^s$ respectively.

E. Optimization and Prediction

We follow the optimization approach in [2], [41] to learn the spectral latent vectors and domain invariant user mapping with RMSprop. The RMSprop is an adaptive version of gradient descent which controls the step size with respect to the

TABLE II: Dataset statistics I

Domain Name	# User	# Item	# Rating
Movies and TV	$2,089$ k	201 k	$4,607$ k
Clothing, Shoes and Jewelry	$3,117$ k	$1,136$ k	$5,748$ k
Apps for Android	$1,324$ k	61 k	$2,638$ k
Amazon Instant Video *	427 k	24 k	584 k

absolute value of the gradient. It is done by scaling the updated value of each weight by a running average of its gradient norm.

For the prediction, we focus on improving the performance on the target domain. We use the spectral representation $\mathbf{V}^u$ and $\mathbf{V}^i$ of users and items respectively in the target domain to make a recommendation. For a specific user u_r, we predict the user's preference over an item i_j as $\mathbf{V}^u(u_r) \cdot \mathbf{V}^i(i_j)$, then we sort the preferences as the ranking list for recommendation.

V. Experiment

In this section, we introduce the dataset first. After that, we discuss the baselines that we compare in this paper. Then we give the experimental settings such as the evaluation metrics. Finally, we present the experiments in details. Through the experiment, we respond to the following research questions:

- **RQ1**: Does the source domain information help to improve the recommendation performance in target domain?
- **RQ2**: Will spectral feature be better in improving the cross-domain recommendation performance?
- **RQ3**: Can the adaptive user mapping help to transfer the information across different domains?

A. Dataset

In this paper, we use the Amazon rating dataset [30], where we find the interactions of users and items. The rating data where a user rates an item scoring from 1 to 5 is from May 1996 - July 2014. The dataset consists of 24 different domains, we present part of the statistics as in Table II. The original dataset is the rating data, we follow the convention in [2], [3] to transform the data into implicit interactions.

Each domain shares a set of common users with other domains. In the experiment, we use the *Amazon Instant Video* dataset as the target domain and the other 23 domains as the source domains.

- **Target Domain**: Amazon Instant Video consists of $583,933$ ratings among $426,922$ users and $23,965$ videos originally. Following the convention [2], [3], we ignore the users with less than 5 interactions, and the final domain has $3,113$ users, $5,860$ items with $22,256$ ratings (connections), and the sparsity is 99.878%.
- **Source Domain**: We use the other datasets as the source domain and part of the statistics of the dataset are illustrated in Table II and Table III. And in the experiment, we compare 23 different source domains and illustrate their contributions to the target domain.

B. Baseline

To answer the previous research questions, we compare our proposed model and methods with some state-of-the-art methods. The major task is defined in Def. 3 which focuses on improving the recommendation performance in the target domain. And we categorize the baseline methods into two groups: (1) **Single domain based methods.** To answer RQ1 we should compare our model with other models that are non-cross-domain, e.g., BPR [38], NCF [3], and SpectralCF [2]. (2) **Cross-domain based methods.** For RQ2, we will investigate the capability of spectral feature in transferring the information across different domains, e.g., CMF [33], CDCF [34], CoNet [9] and our proposed model JSCN. For RQ3, we compare the different version of our proposed model to study the function of the adaptive user mapping. We introduce these methods as followings:

- **BPR** [38]: BPR is a **B**ayesian **P**ersonalized **R**anking based Matrix Factorization method, which introduces a pair-wise loss into the Matrix Factorization to be optimized for ranking [42].
- **NCF** [3]: **N**eural **C**ollaborative **F**iltering applies neural architecture replacing the inner product of latent factors. Thus it can model the non-linear interaction of items and users.
- **SpectralCF** [2]: **Spectral C**ollaborative **F**iltering is the SOTA work to learn the spectral feature of users and items, which is based on the BPR pair-wise loss.
- **CMF** [33]: **C**ollective **M**atrix **F**actorization is a matrix factorization based cross domain rating prediction model. In this paper, we change the rating to 0/1 w.r.t. the implicit interaction of users and items.
- **CDCF** [34]: **C**ross-**D**omain **C**ollaborative **F**iltering method model the user-item interaction as the context feature for the factorization machine. With arbitrary source domain, CDCF can treat them as input feature of users, and learn the latent vectors for both users and items.
- **CoNet** [9]: It is the SOTA deep learning method to learn a shared cross-domain mapping matrix such that the information can be transferred. CoNet enables dual knowledge transferring across domains by introducing cross connections from one base network to another and vice versa. We implement the model with the code published by the author [2].
- **JSCN-α**: **J**oint **S**pectral **C**onvolution **N**etwork is our proposed model to learn a cross-domain recommender system. It is based on graph convolutional network to transfer the spectral feature of users across different domains. This model is a simple version without the adaptive user mapping, only enforcing the spectral vector in different domains to be similar.
- **JSCN-β**: It is the complete version of our proposed model, which includes adaptive user mapping.

C. Experimental Setting

Different from the rating score prediction task, the interaction prediction models in this paper should predict items that are interacted with users in the top ranking list. Thus in the experiment, we utilize the *Recall@K* and *MAP@K* to evaluate

[2]http://home.cse.ust.hk/~ghuac/conet-code_only-hu-cikm18-20181115.zip

the performance of models. We usually have thousands of valid items in a given domain, we use $\mathbf{K} = \{20, 40, 60, 80, 100\}$ to present the performance of models.

For the baseline methods, we select the dimension of latent vectors from $8, 16, 32, 64, 128$ for BPR and SpectralCF. And we follow the suggestion in original papers for NCF to train 3-layer MLP. We implement the CMF model by using the 0-1 interaction matrix. For CDCF, the dimension is set to 32 which is the same as all the cross-domain based model. For our proposed model, there are some hyperparameters requiring tuning. To reduce the complexity of the proposed model, we would let the dimension of invariant user representation equal to the dimension of the spectral latent vector, i.e., $d' = d$. And we set the convolutional dimension parameter $F = C = 32$. The number of filters is important to the performance of the model. And with the validation on different source domain datasets, we find when the number of filters $K = 5$, the performance of JSCN is the best for most of the source domains. We present the validation on JSCN-β with source domain as *Apps for Android* in Figure 4. And we use the linear mapping for domain adaptive part as suggested in Sec. V-G. For the training process, we set the learning rate as 0.001 and the regularization weight ϵ as 0.001.

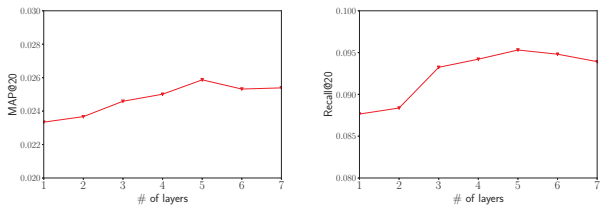


Fig. 4: Validation performance of JSCN-β for the hyper-parameter the number of convolutional layers w.r.t. **MAP@20** and **Recall@20** on target domain.

D. Cross-domain Comparison

To answer **RQ1**, in this experiment part, we would compare the single domain based methods with the cross-domain based models on the same domain. The target domain is the *Amazon Instant Video* dataset. And to answer **RQ2**, we would use the same source domain to compare different cross-domain based methods. To answer the **RQ3**, we would compare the performance of different versions of JSCN, i.e., JSCN-α and JSCN-β. In this section, we would use three different source domain datasets to improve the recommendation performance, which are *Movies and TV*, *Clothing, Shoes and Jewelry* and *Apps for Android*. We analyze the results in details.

In Fig. 5, we present the performance of different models on the target domain w.r.t. *Recall@K*. And in Fig. 6, we show the performance w.r.t. *MAP@K*. For the cross-domain based models, JSCN-β performs the best compared to all the other methods. JSCN-β improves the performance of SpectralCF by 10.2% on recall on average, and 38.3% on MAP on average, which answers that cross-domain information can improve the performance. CMF cannot achieve a good performance compared to the other cross-domain based models. Among all the single domain based models, according to the result in [2] and our results, SpectralCF is the best model compared to NCF and BPR as it can not only model the positive and negative interactions of user-item but also, with the graph convolution, model the interaction in a high-order non-linear way. From the result, some cross-domain based models cannot always surpass the single domain based models.

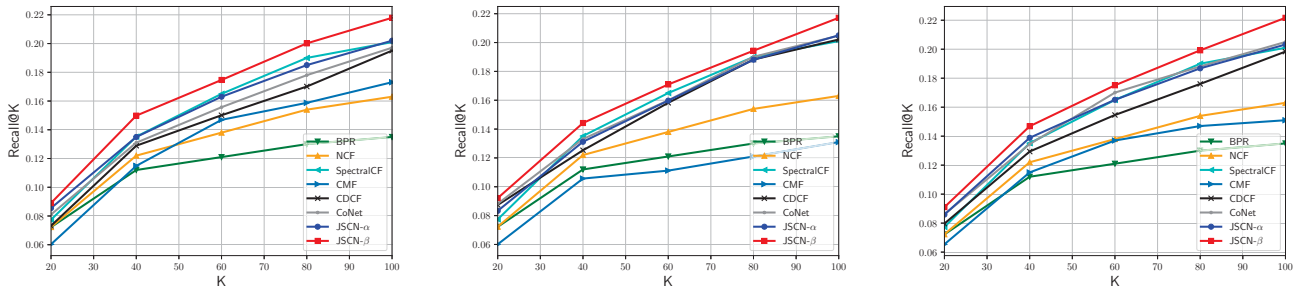
CDCF, CoNet, JSCN-α, and JSCN-β can all well transfer the information across different domains. But since CDCF and CoNet has no spectral convolutional architecture, it cannot capture the high-order interactions of user-item. From our results, SpectralCF can achieve comparable performance with CDCF and CoNet even without source domain information. This suggests that we should apply spectral convolution to transfer the information across different domains. CoNet can transfer the information that learned from the neural networks and shared across different networks. But it cannot capture the high-order information across domain. JSCN-β beats the performance of CoNet by 13.2% on recall in average and 35.2% on MAP in average, which answers that the spectral representation generated by JSCN can improve the performance in cross-domain recommendation.

The users in source domain *Movies and TV* should request similar aspects of items with the users in target domain *Amazon Instant Video* as the items are similar. Thus it is straightforward to transfer the information across these two compatible domains. The result is illustrated in Fig. 5a and Fig. 6a. The performance of JSCN-β and JSCN-α are relatively close. However, the source domain *Clothing, Shoes and Jewelry* is incompatible with the target domain. From the result in Fig 5b and Fig 6b, we can find both JSCN-α and CDCF cannot improve the performance compared to SpectralCF. But JSCN-β learns the domain-invariant user representation which can transfer the information even the domain is incompatible. As a result, the adaptive user mapping in JSCN-β is important to transfer the information across different domains even if the domains are incompatible. JSCN-β beats the performance of JSCN-α by **9.2%** on recall in average and **36.4%** on MAP in average, which answers that the adaptive user mapping can solve the domain incompatible problem thus improve the performance in cross-domain recommendation.

E. Comparison with Different Source Domains

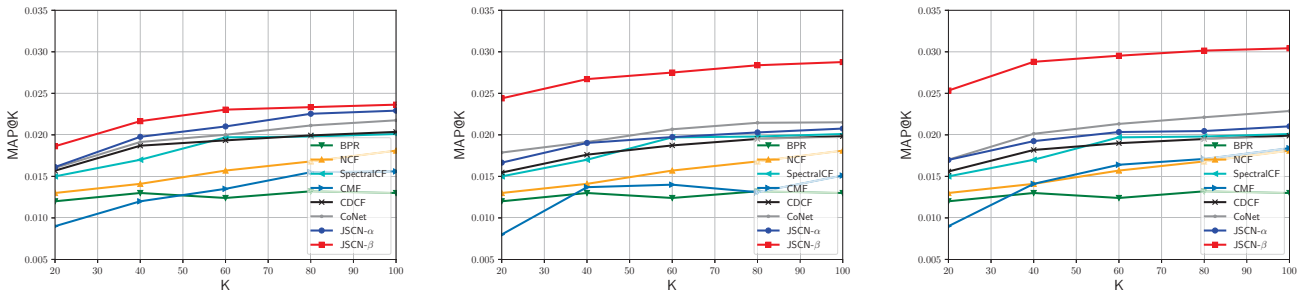
In this section, we report the cross-domain recommendation results of JSCN-β on the target domain with different source domains w.r.t. **MAP@20** in Fig. 7. Since the recall performance varies little for different source domains, and due to the space limitation of the paper, we choose not to show the results of recall.

The best result is from source domain *Apps for Android*. And we can find that even if some of the source domains are incompatible with the target domain *Amazon Instant Video*, e.g. *Clothing, Shoes and Jewelry*, the cross-domain recommendation performs well. Even if some of the source domains



(a) **Movies and TV** (b) **Clothing, Shoes and Jewelry** (c) **Apps for Android**

Fig. 5: Performance comparison w.r.t. **Recall@K** on target domain *Amazon Instant Video*, and with source domain *Movies and TV*,*Clothing, shoes and Jewelry* and *Apps for Android* respectively.



(a) **Movies and TV** (b) **Clothing, Shoes and Jewelry** (c) **Apps for Android**

Fig. 6: Performance comparison w.r.t. **MAP@K** on target domain *Amazon Instant Video*, and with source domain *Movies and TV*,*Clothing, Shoes and Jewelry* and *Apps for Android* respectively.

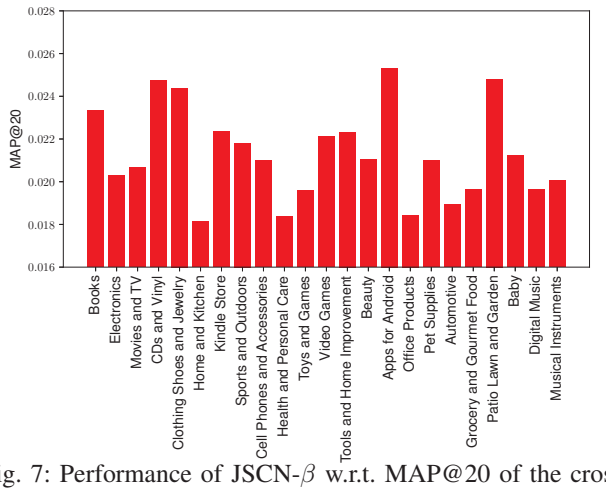


Fig. 7: Performance of JSCN-β w.r.t. MAP@20 of the cross-domain recommendation on target domain.

e.g. *Home and Kitchen*, *Health and Personal Care*, and *Office Products*, perform not that well compared with other source domains, they still improve the performance of SpectralCF by 12.5%, 13.9% and 14.3% respectively, which suggests the benefits of source domain information and the effectiveness of our proposed model.

TABLE III: Dataset Statistics II

label	Domain Name	# User	# Item	# Ratings
1	Home and Kitchen	$2,512$ k	410 k	$4,254$ k
2	Health & Personal Care	$1,851$ k	252 k	$2,982$ k
3	Office Products	909 k	130 k	$1,243$ k

F. Multi-Source JSCN

From the results in Sec. V-E, we notice that our model performs differently given different source domain. Some source domains cannot provide enough information and hence the cross-domain recommendation results are not that good compared to the other source domains. The JSCN model can combine the information from M source domains and share the information together to improve the performance on the target domain. In this section, we conduct the experiment on training JSCN models on multiple source domains.

We select three source domains: *Home and Kitchen*, *Health and Personal Care*, and *Office Products*, which perform worst compared with the other source domains. The domain statistics are summarized in Table III. We conduct the experiment by choosing two out of three source domains to jointly learn the JSCN model. Hence we have $M = 2$ and $M = 3$ in this experiment. The comparison result is presented in Fig. 8.

From the result, when $M = 2$ we can find that multiple source domains can improve the performance compared with single source domain. Especially the combination of *Home and Kitchen* and *Health and Personal Care* source domains

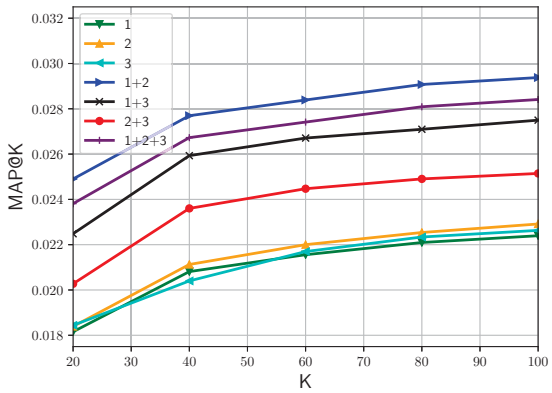


Fig. 8: Perfomrance of JSCN-β w.r.t. MAP@K with different source domains and their combinations. The domain label can be referred in Table III

TABLE IV: The MAP@100 result of variants of JSCN

Source Domain	Books	MT	CSJ	AfA
JSCN-α	0.02374	0.02291	0.02076	0.02103
JSCN-β-MLP	0.02678	**0.02375**	0.02654	0.02537
JSCN-β	**0.02769**	0.02364	**0.02877**	**0.03043**

improve the performance by **37.2%** on average compared using each one of the two domains. This experiment can prove the JSCN can jointly learn the information from multiple source domains. When $M = 3$ we find the performance is a little bit worse than the combination of source domain 1 and source domain 2 (but still better than the other two combinations), which suggests that M also requires tuning. The reason why $M = 2$ is better than $M = 3$ is that the source domain 3 has smaller density value [3] compared with the other 2 domains, which can induce more disturbance to the model.

G. Domain Adaptive Module

In this part, we compare the performance of JSCN-α and JSCN-β with different mapping functions. Recall that JSCN-α is the simple version of the joint spectral convolutional network which enforces the common user latent vector to be similar without the domain adaptive module. As for the domain adaptive module of JSCN-β, we have either the linear mapping or non-linear multi-layer perceptron (MLP) mapping. We use four source domains, i.e. *Books*, *Movies and TV* (MT), *Clothing, Shoes and Jewelry* (CSJ) and *Apps for Android* (AfA).

From the result in Table IV and Table V, we can find JSCN-β performs much better than JSCN-α, which shows the effectiveness of the domain adaptive user mapping module. One interesting observation is the linear mapping beats the non-linear mapping. Since the non-linear mapping requires tuning a lot of hyper-parameters, such as choosing the activation function and the dimension of the hidden layer, we suggest using the linear mapping function for learning the invariant user vector. One possible explanation for this observation is that since the spectral vectors are already low dimensional vectors, MLP can easily find a mapping function such that the invariant user vectors in different domains to be the same, hence over-fitting the user vectors. As over-fitting will harm the structural regularization [40] of the domain adaptive user mapping, the information cannot be transferred in a good way compared with linear mapping.

[3]Density: 1 Home and Kitchen : 0.33%, 2 Health and Personal Care : 0.42% and 3 Office Products : 0.14%

TABLE V: The Recall@100 result of variants of JSCN

Source Domain	Books	MT	CSJ	AfA
JSCN-α	0.2011	0.2021	0.2050	0.2032
JSCN-β-MLP	0.2107	0.2165	0.2112	0.2097
JSCN-β	**0.2187**	**0.2179**	**0.2155**	**0.2217**

VI. Conclusion

In this paper, we design a Joint Spectral Convolutional Network (JSCN) to solve the cross-domain recommendation problem. Firstly, JSCN operates multi-layer spectral convolutions on different graphs simultaneously. Secondly, JSCN maps the learned spectral latent vectors to a domain invariant user representation with adaptive user mapping module. Finally, JSCN minimizes both the in-domain loss in the spectral latent vector space and the cross-domain loss in the domain invariant user representation space to learn the parameters. From the experiment, we can answer three questions: 1)JSCN can use the source domain information to improve the recommendation performance; 2) the spectral convolutions in JSCN can capture the comprehensive connectivity information to improve the performance in cross-domain recommendation; 3) the adaptive user mapping of learning the domain-invariant representation can help to transfer knowledge across different domains.

VII. Acknowledgement

This work is supported in part by NSF under grants III-1526499, III-1763325, III-1909323, CNS-1930941, and CNS-1626432. This work is also partially supported by NSF through grant IIS-1763365 and by FSU.

REFERENCES

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[2] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018, pp. 311–319.

[3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[4] T. Man, H. Shen, X. Jin, and X. Cheng, "Cross-domain recommendation: an embedding and mapping approach," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 2464–2470.

[5] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *WWW*. International World Wide Web Conferences Steering Committee, 2015, pp. 278–288.

[6] A. Farseev, I. Samborskii, A. Filchenkov, and T.-S. Chua, "Cross-domain recommendation via clustering on multi-layer graphs," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 195–204.

[7] J. Zhang and P. S. Yu, *Broad Learning Through Fusions - An Application on Social Networks*. Springer, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-12528-8

[8] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and C. Zhu, "Personalized recommendation via cross-domain triadic factorization," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 595–606.

[9] G. Hu, Y. Zhang, and Q. Yang, "Conet: Collaborative cross networks for cross-domain recommendation," in *CIKM*. ACM, 2018, pp. 667–676.

[10] X. Wang, Z. Peng, S. Wang, S. Y. Philip, W. Fu, and X. Hong, "Cross-domain recommendation for cold-start users via neighborhood based feature mapping," in *International Conference on Database Systems for Advanced Applications*. Springer, 2018, pp. 158–165.

[11] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[12] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016, pp. 353–362.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[14] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.

[15] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[16] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[17] Y. Shi, Q. Zhu, F. Guo, C. Zhang, and J. Han, "Easing embedding learning by comprehensive transcription of heterogeneous information networks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2190–2199.

[18] M. Kazama and I. Varga, "Cross domain recommendation using vector space transfer learning." in *RecSys Posters*, 2016.

[19] C.-Y. Li and S.-D. Lin, "Matching users and items across domains to improve the recommendation quality," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 801–810.

[20] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.

[21] Y. Zheng, B. Tang, W. Ding, and H. Zhou, "A neural autoregressive approach to collaborative filtering," *arXiv preprint arXiv:1605.09477*, 2016.

[22] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[23] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[24] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 627–636.

[25] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 233–240.

[26] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 425–434.

[27] R. He and J. McAuley, "Vbpr: Visual bayesian personalized ranking from implicit feedback." in *AAAI*, 2016, pp. 144–150.

[28] Q. Liu, S. Wu, and L. Wang, "Deepstyle: Learning user preferences for visual recommendation," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 841–844.

[29] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 43–52.

[30] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 507–517.

[31] Y. Zhang, Q. Ai, X. Chen, and W. Croft, "Joint representation learning for top-n recommendation with heterogeneous information sources," *CIKM. ACM*, 2017.

[32] S. Zhang, L. Yao, and A. Sun, "Deep learning based recommender system: A survey and new perspectives," *arXiv preprint arXiv:1707.07435*, 2017.

[33] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 650–658.

[34] B. Loni, Y. Shi, M. Larson, and A. Hanjalic, "Cross-domain collaborative filtering with factorization machines," in *European conference on information retrieval*. Springer, 2014, pp. 656–661.

[35] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[36] G. Hu, Y. Zhang, and Q. Yang, "Mtnet: a neural approach for cross-domain recommendation with unstructured text." KDD, 2018.

[37] J. Liu, P. Zhao, Y. Liu, V. S. Sheng, F. Zhuang, J. Xu, X. Zhou, and H. Xiong, "Deep cross networks with aesthetic preference for cross-domain recommendation," *arXiv preprint arXiv:1905.13030*, 2019.

[38] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 452–461.

[39] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[40] J. Zhou, J. Chen, and J. Ye, "Malsar: Multi-task learning via structural regularization," *Arizona State University*, vol. 21, 2011.

[41] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent."

[42] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Mymedialite: A free recommender system library," in *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 2011, pp. 305–308.