Transfer Learning in Cross-Domain Sequential Recommendation

Zitao Xu

Abstract

Sequential recommendation captures users' dynamic preferences by modeling the sequential information of their behaviors. However, most existing works only focus on users' behavior sequences in a single domain, and suffer from the data sparsity issue. We notice that there are also item transition patterns across sequences from different domains, which means that a user's interaction in one domain may affect his/her interaction in the other domains at the next time. In this paper, we aim to improve the performance of sequential recommendation in the target domain by introducing users' behavior sequences from multiple source domains, and propose a novel solution named transfer via joint attentive preference learning (TJAPL). Specifically, we tackle the studied problem from the perspective of transfer learning. We adopt the self-attention sequential recommendation (SASRec) model for target-domain attentive preference learning (APL) to capture the users' dynamic preferences. Furthermore, we design cross-domain user APL to transfer and share the users' overall preferences from the source domains to the target domain, leveraging the behavior sequences from the source domains to address the scarcity challenge. We also design cross-domain local APL to capture the transition patterns across different domains for knowledge transfer. Notice that our TJAPL can be applied to scenarios with multiple source domains. Extensive empirical studies indicate that our TJAPL significantly outperforms ten state-of-the-art baselines.

Keywords: Cross-Domain Recommendation, Sequential Recommendation, Attentive Learning, Transfer Learning

1 Introduction

Sequential recommendation aims to predict the next item that a user is most likely to interact with based on his/her behavior sequence. It has become increasingly important because of its great accuracy and practicability by capturing the sequential information. Existing studies often adopt recurrent neural networks (RNNs) and attention mechanisms to model sequential patterns from user historical interactions[1-4]. However, most of studies only focus on a user' s behavior sequence in one single domain and suffer from the cold-start and data sparsity problem commonly existed in recommender systems.

Cross-domain recommendation has been proposed to leverage the relatively richer information from some source domains to improve the recommendation performance in a target domain, which is useful to alleviate the data sparsity problem[5]. However, it often does not take into account the sequential information of user behaviors.

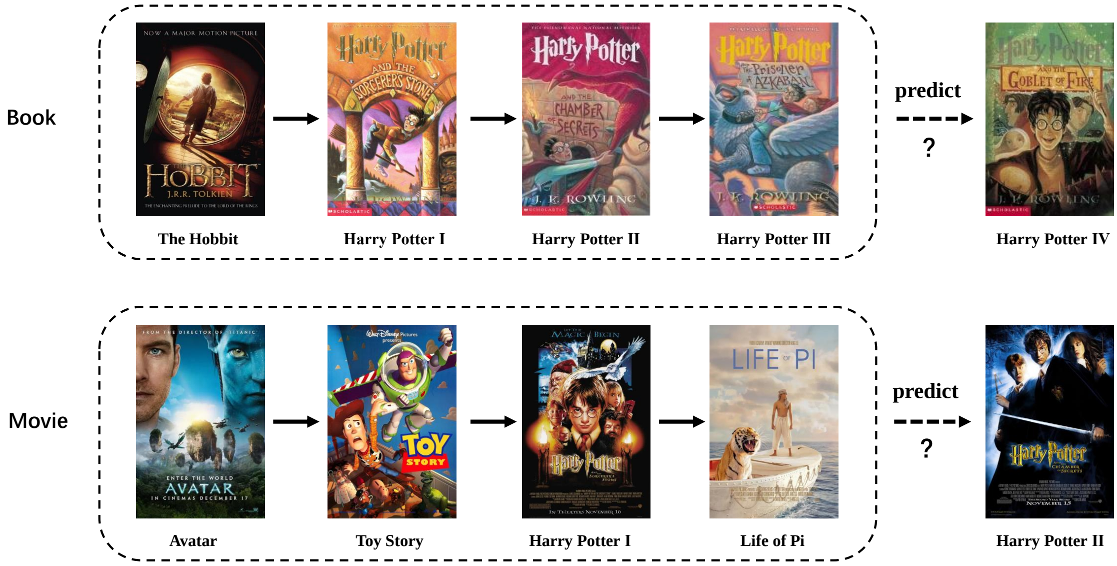


Figure 1: An example of a user's behavior sequences in the book domain (the top half) and the movie domain (the bottom half).

In this paper, we study a new and emerging problem, i.e., cross-domain sequential recommendation. We aim to combine the sequential information with some other domain data, which not only captures the dynamic preferences from users' behavior sequences, but also effectively alleviates the data scarsity issue that usually occurs in a single domain. Moreover, we notice that the items a user interacts with over time may come from multiple domains, due to the diversity of users' preferences. There are also item transition patterns across sequences from different domains, which implies that a user's interaction in one domain may affect his/her next interaction in another domain.

Scenarios of cross-domain sequential recommendation are common in real-world situations. For example, from the top part of Fig. 1, we can see that the user has recently read three "Harry Potter" (the books), and following the idea of sequential recommendation, we will recommend the sequel of "Harry Potter" (the book) for him/her, which obviously has a higher probability of satisfying his/her needs. From the bottom half of Fig. 1, we can see that in the movie domain, the user recently tends to watch fantasy movies, but due to the uncertainty of the user's intention, there is no strong connection between these movies. If we account for the user's recent behaviors in the book domain, we can find that he/she is interested in "Harry Potter" (the books), so it may be a good suggestion to recommend "Harry Potter" (the movie) for him/her. It can be seen that if both the sequential information and the cross-domain behaviors are considered, the recommendation performance will be improved.

There are two main challenges for cross-domain sequential recommendation: i) how to address the data scarcity problem on users' behavior sequences in the target domain, and ii) how to extract the correlation between users' preferences in the target domain and source domains. There are relatively few existing works on cross-domain sequential recommendation, and most methods are based on RNNs[6-7], which have limited capability to capture the complex associations between domains and are also difficult to be parallelized. Moreover, these methods often focus on the associations of one single source domain to a certain target domain, while transferring knowledge from multiple domains may be more helpful in practical applications.

2 Related works

In this section, we briefly describe the related works from four categories: (i) general recommendation, (ii) cross-domain general recommendation, (iii) sequential recommendation, and (iv) cross-domain sequential recommendation.

2.1 General Recommendation

In early works of recommender systems, users' preferences are typically modeled by using collaborative filtering (CF) methods. Matrix factorization (MF) based methods[8-9] is one main branch of CF methods. Specifically, it projects users and items into a shared vector space and then predicts a user's rating on an item by the inner product of the two corresponding vectors. Another line of work is neighborhood-based methods[10-11], which try to make recommendations based on (item, item) or (user, user) similarities. Recently, deep learning (DL) based methods are adopted to improve the recommendation performance. For example, neural collaborative filtering (NCF)[12] uses multilayer perceptron (MLP) to learn user preferences while CVAE[13] uses an autoencoder (AE) to predict users' ratings.

2.2 Cross-Domain General Recommendation

Cross-domain recommendation is proposed to solve the data sparsity problem and the cold-start issue in a typical domain. It improves the recommendation performance in a target domain by transferring knowledge from a source domain. In cross-domain recommendation, the most important concern is determining what knowledge to transfer between domains and how to transfer the knowledge. A representative branch to transferring knowledge is the mapping-based methods, which model the connection between two domains by explicitly learning a mapping function. For example, EMCDR[14] learns users' preferences in the source and target domains, respectively, and then transfers the overlapped users' preferences in the source domain to the target domain by a mapping function. DDTCDR[15] introduces a latent orthogonal mapping to capture user preferences over multiple domains while preserving relations between users across different latent spaces. Another approach to transferring knowledge is based on multi-domain collaborative training. For example, CMF[16] factorizes matrices from multiple domains simultaneously and enables knowledge transfer by sharing the user's latent factors. CoNet[17] develops a collaborative cross-network to allow dual knowledge transfer across different domains.

2.3 Sequential Recommendation

Unfortunately, the methods mentioned in Section 2.1 and Section 2.2 are not suitable for sequential recommendation since they all ignore the order in users' behaviors. Early works on sequential recommendation are based on Markov chains (MCs) aiming to capture sequential patterns among successive items. For instance, factorized personalized MCs (FPMC)[18] combine MCs and MF[9] to model short-term preference and long-

Table 1: A summary of some related problems, i.e., general recommendation, cross-domain general recommendation, sequential recommendation and cross-domain sequential recommendation.

	Single-Domain	Cross-Domain
Non-Sequential	FISM[32], etc. (General recommendation, many)	CoNet[17], etc. (Cross-domain general recommendation, many)
Sequential	SASRec[3], etc. (Sequential recommendation, many)	π-Net[6], DA-GCN[33] **(Cross-domain sequential recommendation, few)**

term preference, respectively.[19] adopt high-order MCs to consider more previous items. Recently, recurrent neural networks (RNNs) are introduced to sequential recommendation since their natural instincts to model sequence step-by-step[1-2]. Caser[20] proposes a CNN-based method which adopts both horizontal and vertical convolutional filters to learn sequential patterns. Moreover, multi-head attention mechanisms[21] are employed to sequential recommendation which can avoid the vanishing gradient problem brought by RNNs[3-4,22]. SASRec[3] uses self-attention blocks to model users' behavior sequences, which is proved to be effective and efficient. Graph neural networks (GNN)[23-24] are adopted to extract features with more consistency or adjacency consideration. There are also some works[22,25-26] focusing on capturing the user's overall preference or global representation to generate the user's general interests. Although these studies have made great progress, none of them has considered introducing some source-domain data or transferring knowledge under cross-domain situations.

2.4 Cross-Domain Sequential Recommendation

Recently, π-net[6] and its improved version[7] are proposed for cross-domain sequential recommendation in a shared-account scenario. Specifically, π-net devises a cross-domain transfer unit to extract and share user information between two domains at each timestamp.[27] employs knowledge graphs to improve knowledge transfer across domains. However, these methods are based on RNNs, which may not be expressive enough to capture the multiple associations between domains and make models less efficient. More recently, cross-domain sequential recommendation is likely to emerge as a powerful approach to tackle the cold-start and sparsity problem for click-through-rate (CTR) prediction in online commerce platforms[28-29]. CD-SASRec[30] proposes an improved method based on SASRec[3] that fuses the source-domain aggregated vector into the target-domain item embedding to transfer information across domains, but it neglects to explore the sequential information across domains. Moreover, most of the studies consider using only one source domain to improve the recommendation performance in a target domain, and cannot be directly applied to multiple domains.

In this work, we base our target-domain attentive preference learning module on SASRec[3], which has been shown as an effective and efficient model in various previous works on sequential recommendation[25-26,31]. We aim to improve SASRec by transferring a user's overall preferences and sequential information from multiple source domains (rather than from one single domain) to a target domain.

Table 2: Important notations and their explanations.

Symbol	Explanation
$\mathcal{U}$	user set
$\mathcal{I}$	item set for the target domain
N	number of source domains
$\mathcal{I}^{S_n}$	S_n represents the n-th source domain; $\mathcal{I}^{S_n}$ is the item set for the n-th source domain
u	user $u \in \mathcal{U}$
v_i	the item that user interacted with at time step i in the target domain
$v_j^{S_n}$	the item that user interacted with at time step j in the n-th source domain
L	maximum sequence length
B	number of attention blocks
$\mathcal{V} = \{v_1, v_2, \dots, v_L\}$	user's interaction sequence in the target domain
$\mathcal{V}_t = \{v_1, v_2, \dots, v_t\}$	truncated item sequence at time step t with regard to sequence $\mathcal{V}$
$\mathcal{V}_{t'}^{S_n} = \{v_1^{S_n}, v_2^{S_n}, \dots, v_{t'}^{S_n}\}$	truncated item sequence at time step t' for the n-th source domain
d	latent vector dimensionality
$\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{V}^{S_n}$	embedding associate with u, $\mathcal{V}_t$, $\mathcal{V}_{t'}^{S_n}$
$\boldsymbol{p}_t$	position embedding at time step t
$\boldsymbol{f}_t$	target-domain attentive preference at time step t
$\boldsymbol{f}_t^u$	cross-domain user attentive preference at time step t
$\boldsymbol{f}_t^{S_n}$	cross-domain local attentive preference at time step t
$\boldsymbol{o}_t$	final representation of the user's preference at time step t
$r_{t,i}$	preference score of item i at time step t

3 Method

In this section, we first formalize the studied task, i.e., cross-domain sequential recommendation. Then, we introduce the detailed components of our model, i.e., transfer via joint attentive preference learning (TJAPL). As shown in Fig. 2, our TJAPL mainly consists of a target-domain attentive preference learning module (TD-APL), a cross-domain user attentive preference learning module (CD-UAPL), and a cross-domain local attentive preference learning module (CD-LAPL). For ease of reading and understanding, we summarize the key notations and their explanations in Table 2.

3.1 Problem Definition

For cross-domain sequential recommendation, we have the set of users $\mathcal{U}$, and we denote the set of items in the target domain as $\mathcal{I}$. Moreover, we have N source domains with same users and different item sets $\mathcal{I}^{S_n}$, $1 \leqslant n \leqslant N$. We define the target-domain behavior sequence of each user $u \in \mathcal{U}$ as $\mathcal{V} = \{v_1, v_2, \dots, v_L\}$ (ordered by the interaction time), which consists of L items from $\mathcal{I}$. And we will repeatedly append a padding item at the beginning of the sequence if the sequence length is shorter than L. Moreover, $\mathcal{V}_t = \{v_1, v_2, \dots, v_t\}$, $1 \leqslant t \leqslant L$ denotes a truncated behavior sequence at time step t with regard to sequence $\mathcal{V}$. Specifically, for the n-th source domain, we denote a truncated item sequence as $\mathcal{V}_{t'}^{S_n} = \{v_1^{S_n}, v_2^{S_n}, \dots, v_{t'}^{S_n}\}$, where t' is the most recent time step at which the user interacted with an item in the n-th source domain before the real moment corresponding to the time step t in the target domain. This is to ensure causality of the user behaviors from the

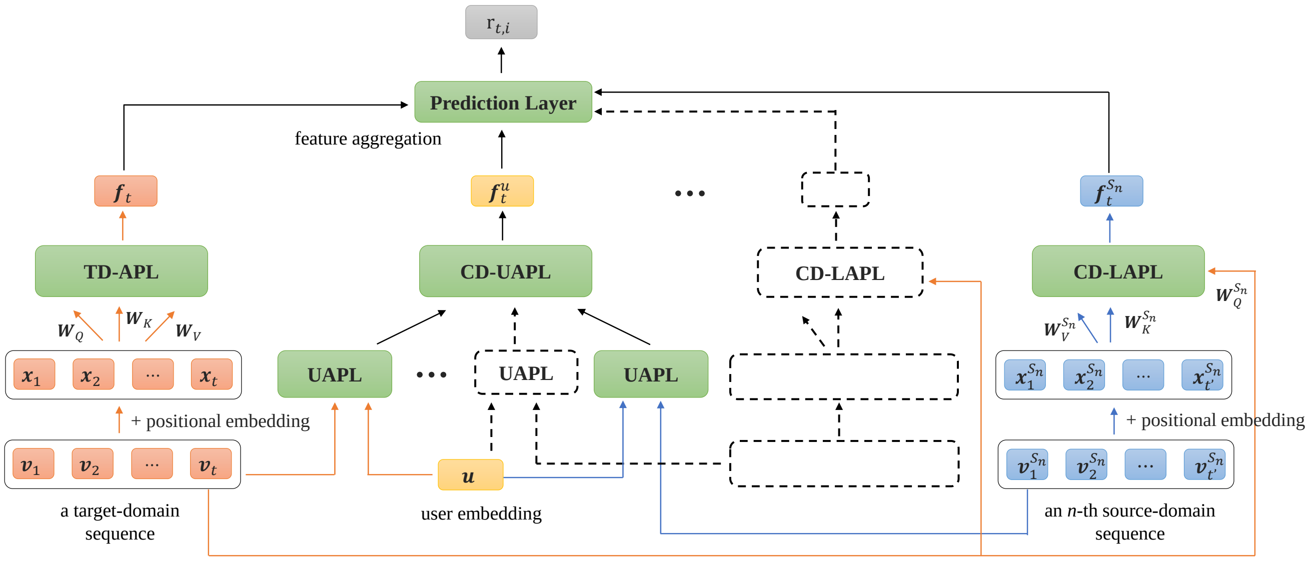


Figure 2: The framework of our proposed TJAPL (transfer via joint attentive preference learning). TD-APL (target-domain APL) is fed with the embedding of a target domain sequence, which contains some self-attention blocks (see Eqs.(3∼7)). CD-UAPL (cross-domain user APL) extracts a user's overall preference in all domains, where each domain includes a user attention layer (see Eqs.(8∼11))) to capture the user preference in the corresponding domain. CD-LAPL (cross-domain local APL) is fed with the embedding of a target-domain sequence and a source-domain sequence which consists of cross-domain attention blocks (see Eqs.(14∼16)). Notice that each source domain contains its own CD-LAPL.

n-th source domain to the target domain. Cross-domain sequential recommendation aims to predict the next likely to be preferred item in the target domain (i.e., v_{t+1}) according to $\mathcal{V}_t$ and $\mathcal{V}_{t'}^{S_n}$ where $1 \leqslant n \leqslant N$. The left part of Fig. 2 is the input sequence of the target domain, and the right part is that of the n-th source domain.

3.2 Target-Domain Attentive Preference Learning

We first denote $\boldsymbol{u}$ as the embedding vector of the user u in $\mathcal{U}$, where $\boldsymbol{u} \in \mathbb{R}^d$ is a learnable vector. Similarly, we denote $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_t\}$ as the embedding of the target-domain sequence $\mathcal{V}_t$ and $\boldsymbol{V}^{S_n} = \{\boldsymbol{v}_1^{S_n}, \boldsymbol{v}_2^{S_n}, \ldots, \boldsymbol{v}_{t'}^{S_n}\}$ as the embedding of the n-th source-domain sequence $\mathcal{V}_{t'}^{S_n}$.

We employ attention mechanism[3,21] to explore the sequential patterns in the target domain. Since the self-attention model can't consider the positions of the previous items, a learnable position embedding $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_L\} \in \mathbb{R}^{L\times d}$ should be added to the sequence embedding $\boldsymbol{V}$ and $\boldsymbol{V}^{S_n}$, then we obtain the position-aware input embedding $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$ and $\boldsymbol{X}^{S_n} = \{\boldsymbol{x}_1^{S_n}, \boldsymbol{x}_2^{S_n}, \ldots, \boldsymbol{x}_{t'}^{S_n}\}$,

$$\boldsymbol{x}_i = \boldsymbol{v}_i + \boldsymbol{p}_i, , \tag{1}$$

$$\boldsymbol{x}_i^{S_n} = \boldsymbol{v}_i^{S_n} + \boldsymbol{p}_i. \tag{2}$$

Next, we feed the sequence $\boldsymbol{X}$ into some stacked self-attention blocks (SABs), which is the same with that in[3]. Omitting the residual connection layers and the normalization layers, each SAB is regarded as a self-attention layer $SAL(\cdot)$ followed by a feed-forward network $FFN(\cdot)$. Specifically, $SAL(\boldsymbol{X})$ can be formalized as:

$$\alpha_i = softmax\left(\boldsymbol{x}_t\boldsymbol{W}_Q\left(\boldsymbol{x}_i\boldsymbol{W}_K\right)^T\right), \forall i \in \{1, 2, \ldots, t\}, \tag{3}$$

$$\boldsymbol{h}_t = \sum_{i=1}^{t} \alpha_i \left(\boldsymbol{x}_i\boldsymbol{W}_V\right), \tag{4}$$

where $\boldsymbol{x}_t\boldsymbol{W}_Q$, $\boldsymbol{x}_i\boldsymbol{W}_K$, and $\boldsymbol{x}_i\boldsymbol{W}_V$ stand for Query, Key, and Value, respectively. $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V \in \mathbb{R}^{d\times d}$ are learnable parameters that improve the flexibility of the model. More clearly, the importance of Value is measured by using Query to match against Key. In this case, it refers to using the item which was interacted with at the last time step to match those items he/she interacted with before, then obtain the item weighting information to generate the information used for prediction at the next time step, i.e., $\boldsymbol{h}_t \in \mathbb{R}^d$.

Then, we employ a two-layer $FFN(\boldsymbol{h}_t)$ to enable the model to explore the nonlinear features:

$$\boldsymbol{f}_t = ReLU\left(\boldsymbol{h}_t\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}, \tag{5}$$

where $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)} \in \mathbb{R}^d$ are learnable parameters for the two-layer FFN. We utilize the same dropout and normalization layers as in[21] in this module.

Stacking the SAB is usually helpful for the model to extract the more complex sequential patterns. We denote the b-th $(b > 1)$ SAB as:

$$\boldsymbol{h}_t^{(b)} = SAL(\boldsymbol{f}_t^{(b-1)}), \tag{6}$$

$$\boldsymbol{f}_t^{(b)} = FFN(\boldsymbol{h}_t^{(b)}). \tag{7}$$

Finally, we take the final output vector $\boldsymbol{f}_t^{(b)} \in \mathbb{R}^d$ from the top SAB as the target-domain attentive preference, which represents the current interests of user at time step t in the target domain. In the remainder of this paper, we use $\boldsymbol{f}_t$ to denote $\boldsymbol{f}_t^{(b)}$ for simplicity. Furthermore, in contrast to RNNs, the computation of self-attention mechanism can be effectively parallelized.

3.3 Cross-Domain User Attentive Preference Learning

Although TD-APL can capture the behavioral preference from the target-domain sequence, due to the property of the self-attention mechanism, it will rely on the last interaction in a sequence to generate the relevant output. This is not suitable for capturing the user's overall preference and making personalized recommendations. In addition, so far, we have focused only on the target-domain sequential information of users, and how to make use of the source-domain sequences is also one of the issues to be considered.

Inspired by the existing works on single-domain sequential recommendation that devotes to identifying the long-term preferences of users to generate the user' s general interests.[22,25,32], we propose a novel CD-UAPL module on cross-domain sequential recommendation.

According to the attention mechanism introduced in Section 3.2, an effective approach is to take the learnable vector $\boldsymbol{u} \in \mathbb{R}^d$ (i.e., the embedding of user u) as the query in the attention layer, which means that the query is the same for the user u regardless of which time step t the current interaction is at. This is also beneficial for personalized recommendation, because each user has his/her own embedding vector. The target-domain user attentive preference can then be formalized as follows:

$$\beta_i = softmax\left(\boldsymbol{u}\boldsymbol{W}_{Q_u}\left(\boldsymbol{v}_i\boldsymbol{W}_{K_u}\right)^T\right), \forall i \in \{1,2,\ldots,t\}, \tag{8}$$

$$\boldsymbol{z}_t = \sum_{i=1}^{t} \beta_i\left(\boldsymbol{v}_i\boldsymbol{W}_{V_u}\right), \tag{9}$$

where $\boldsymbol{v}_i$ is the initial embedding of item i, $\boldsymbol{W}_{Q_u}$, $\boldsymbol{W}_{K_u}$, $\boldsymbol{W}_{V_u} \in \mathbb{R}^{d\times d}$ are the learnable parameters similar to $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$ in Eq.(3), and $\boldsymbol{u}\boldsymbol{W}_{Q_u}$ denotes the Query. $\boldsymbol{z}_t$ is the target-domain user attentive preference, which stands for the overall preference of user u up to time step t in the target domain.

Notice that we abandon the position information $\boldsymbol{P}$ which is also the difference between Eq.(8) and Eq.(3) in the attention layer besides the query condition, because the long-term preference is not sensitive to the position information of the interactions.

Considering that a same user usually has similar preferences beneath his or her behaviors in different domains, e.g., in the book domain, the user prefers to read fantasy novels, and to watch fantasy movies in the movie domain (as shown in Fig. 1), then the items of the fantasy genre is often more attractive to him. Even though the types of items are different, they reflect the same user preference.

In addition, when the user's behavior in the target domain is highly sparse, using only his or her target-domain interactions may not capture the user's overall preference well. However, if we can additionally generate the user's overall preference from some dense source domains with more interactions, the recommendation performance will be improved. Hence, we formalize the user attentive preference in the n-th source domain as follows:

$$\beta_i^{S_n} = softmax\left(\boldsymbol{u}\boldsymbol{W}_{Q_u}^{S_n}\left(\boldsymbol{v}_i^{S_n}\boldsymbol{W}_{K_u}^{S_n}\right)^T\right), \forall i \in \{1,2,\ldots,t'\}, \tag{10}$$

$$\boldsymbol{z}_t^{S_n} = \sum_{i=1}^{t'} \beta_i^{S_n}\left(\boldsymbol{v}_i^{S_n}\boldsymbol{W}_{V_u}^{S_n}\right), \tag{11}$$

where $\boldsymbol{v}_i^{S_n}$ is the initial input embedding of the n-th source-domain sequence, $\boldsymbol{W}_{Q_u}^{S_n}$, $\boldsymbol{W}_{K_u}^{S_n}$, $\boldsymbol{W}_{V_u}^{S_n} \in \mathbb{R}^{d\times d}$ are learnable parameters, and $\boldsymbol{z}_t^{S_n}$ denotes the user attentive preference in the n-th source domain.

We employ concatenation to aggregate all user attentive preference from different domains, and then feed the concatenation vector into MLP to get the final representation of cross-domain user attentive preference:

$$\boldsymbol{z} = concat\left[\boldsymbol{z}_t, \ldots, \boldsymbol{z}_t^{S_N}\right], \tag{12}$$

$$\boldsymbol{f}_t^u = \boldsymbol{z}\boldsymbol{W}^{(u)} + \boldsymbol{b}^{(u)}, \tag{13}$$

where N denotes the number of source domains, $\boldsymbol{z} \in \mathbb{R}^{(1+N)d}$ denotes the concatenation of all user preferences and $\boldsymbol{W}^{(u)} \in \mathbb{R}^{(1+N)d\times d}$, $\boldsymbol{b}^{(u)} \in \mathbb{R}^d$ are learnable parameters. We take the final output vector $\boldsymbol{f}_t^u \in \mathbb{R}^d$ as the cross-domain user attentive preference.

3.4 Cross-Domain Local Attentive Preference Learning

Considering that the next interaction of user in the target domain may be related to an item he/she recently interacted with in a certain source domain, we propose CD-LAPL to exploit the user's sequential information in multiple domains then transfer knowledge across different domains.

We adapt the attention block which is introduced in Section 3.2 to measure the importance of a user's previous interactions in a source domain to current interaction in the target domain, and explore the transition patterns across sequences from different domains. Specifically, we denote the input embedding of the target-domain item at the last time step $\boldsymbol{v}_t$ as query, and denote the position-aware input embedding of the n-th source-domain sequence $\boldsymbol{X}^{S_n}$ as key and value, and then the cross-domain attention layer can be formalized as follows:

$$\alpha_i^{S_n} = softmax\left(\boldsymbol{v}_t \boldsymbol{W}_Q^{S_n}\left(\boldsymbol{x}_i^{S_n}\boldsymbol{W}_K^{S_n}\right)^T\right), \forall i \in \{1, 2, \ldots, t'\}, \tag{14}$$

$$\boldsymbol{h}_t^{S_n} = \sum_{i=1}^{t'} \alpha_i^{S_n}\left(\boldsymbol{x}_i^{S_n}\boldsymbol{W}_V^{S_n}\right), \tag{15}$$

where $\boldsymbol{W}_Q^{S_n}$, $\boldsymbol{W}_K^{S_n}$, $\boldsymbol{W}_V^{S_n} \in \mathbb{R}^{d\times d}$ are learnable parameters. We also employ a two-layer $FFN(\cdot)$ to further improving the model performance:

$$\boldsymbol{f}_t^{S_n} = ReLU\left(\boldsymbol{h}_t^{S_n}\boldsymbol{W}^{S_n(1)} + \boldsymbol{b}^{S_n(1)}\right)\boldsymbol{W}^{S_n(2)} + \boldsymbol{b}^{S_n(2)}, \tag{16}$$

where $\boldsymbol{W}^{S_n(1)}, \boldsymbol{W}^{S_n(2)} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}^{S_n(1)}, \boldsymbol{b}^{S_n(2)} \in \mathbb{R}^d$ are weights and biases for the two-layer FFN, respectively. And each cross-domain attention block can also be regarded as a cross-domain attention layer followed by a FFN.

We take the top cross-domain attention block's output vector $\boldsymbol{f}_t^{S_n} \in \mathbb{R}^d$ as the cross-domain local attentive preference, which represents a user's cross-domain dynamic interests at the t-th time step reflected from the target domain and the n-th source domain. Notice that for N source domains, we will obtain N cross-domain local attentive preferences.

3.5 Prediction Layer

To combine all the output vectors from TD-APL, CD-UAPL and CD-LAPL, we try different designs for feature aggregation such as concatenation, summation and maximum. In this paper, we employ concatenation to aggregate all features which is the optimal choice as we found in the empirical studies.

$$\boldsymbol{o} = concat\left[\boldsymbol{f}_t, \boldsymbol{f}_t^u, \ldots, \boldsymbol{f}_t^{S_N}\right], \tag{17}$$

where $\boldsymbol{o} \in \mathbb{R}^{(2+N)d}$ denotes the concatenation of all the output vectors. Then, the concatenation vector is fed into an MLP to obtain the final representation of the user's preference:

$$\boldsymbol{o}_t = \boldsymbol{o}\boldsymbol{W}^{(o)} + \boldsymbol{b}^{(o)},, \tag{18}$$

where $\boldsymbol{W}^{(o)} \in \mathbb{R}^{(2+N)d\times d}$ and $\boldsymbol{b}^{(o)} \in \mathbb{R}^d$ are learnable parameters, and $\boldsymbol{o}_t \in \mathbb{R}^d$ denotes the final representation of the user's preference. Finally, the prediction score of item i can be calculated as follows:

$$r_{t,i} = \boldsymbol{o}_t(\boldsymbol{v}_i)^T., \tag{19}$$

We adopt Adam as the optimizer and the binary cross-entropy loss function for our TJAPL can be formalized as:

$$\mathcal{L} = -\sum_{u\in\mathcal{U}}\sum_{t=1}^{L-1}\delta(v_{t+1})[\log(\sigma(r_{t,v_{t+1}}) + \log(1-\sigma(r_{t,j})], , \tag{20}$$

where $j \in \mathcal{I}\backslash\mathcal{V}^u$ is a sampled negative item and σ is the sigmoid function. The indicator function $\delta(v_{t+1}) = 1$ only if v_{t+1} is not a padding item, and 0 otherwise.

4 Implementation details

4.1 Comparing with released source codes

SASRec[3] is a seminar method which employs the attention mechanism to sequential recommendation. It is more efficient and has an advantage in capturing the long-range dependency compared with RNN-based models, but it cannot handle the cross-domain scenarios. We implement SASRec[1] following the published codes of the original papers.

In order to deal with the cross-domain sequential recommendation task, we propose a improved model based on SASRec. We treat the SASRec model as TD-APL to capture the dynamic preference in the target domain. Furthermore, We believe that there is also some important sequential information in the source domain, because the next interaction of user in the target domain is probably related to an item he/she recently interacted with in a certain source domain. Therefore, we not only design CD-UAPL to capture a user's overall preference, but also employ CD-LAPL to explore the transition patterns across sequences from different domains. Notice that capturing sequential information of users is also the key to distinguish cross-domain sequential recommendation from cross-domain general recommendation. Moreover, our TJAPL can be applied to scenarios with multiple source domains, while transferring knowledge from multiple domains may be more helpful in practical applications.

4.2 Experimental environment setup

We implement our TJAPL with an adaption code from the published code of SASRec. We mainly run our code in 'python 3.7+, PyTorch 1.6'. We adopt Adam as the optimizer[34] and train our model on a single Tesla V100 PCIe GPU. The detail of the datasets and hyper-parameters will be disscussed in Session 5.1 and 5.4.

4.3 Main contributions

In this paper, we study a new and emerging problem, i.e., cross-domain sequential recommendation, and we propose a novel solution named transfer via joint attentive preference learning (TJAPL). Specifically, we tackle the studied problem from the perspective of transfer learning and attentive preference learning (APL). Our TJAPL contains target-domain APL (TD-APL) and cross-domain APL, where the latter is further divided into cross-domain user APL (CD-UAPL) and cross-domain local APL (CD-LAPL). In particular, we treat the self-attention sequential recommendation (SASRec) model[3] as TD-APL to model the users' behavior sequences and capture their dynamic preferences in the target domain. Furthermore, considering that users may have similar interests in multiple domains, we propose CD-UAPL to share and transfer the users' overall pref-

[1]https://github.com/kang205/SASRec

Algorithm 1: The learning procedure of transfer via joint attentive preference learning (TJAPL)

1: **Initialization:** Initialize model parameters Θ.
2: **repeat**
3:   **for** *each epoch* **do**
4:     Collect a batch of users and their corresponding sequences in the target domain and the source domains.
5:     Calculate the target-domain attentive preference $\boldsymbol{f}_t$ of time step t via Equations (1 - 7).
6:     Calculate the cross-domain user attentive preference $\boldsymbol{f}_t^u$ of time step t via Equations (8 - 13).
7:     **for** $n \leftarrow 1$ **to** N **do**
8:       Calculate the cross-domain local attentive preference $\boldsymbol{f}_t^{S_n}$ of time step t via Equations (14 - 16).
9:     **end for**
10:     Calculate the final representation of the user's preference $\boldsymbol{o}_t$ of time step t via Equations (17 - 18).
11:     Predict the preference score $r_{t,i}$ of item i at each time step t via Equation (19).
12:     Calculate the binary cross-entropy loss $\mathcal{L}$ via Equation (20).
13:     Update the model parameters via $\nabla_\Theta \mathcal{L}$.
14:   **end for**
15: **until** Convergence

erences from more than one source domains to the target domain, leveraging the sequential behaviors from the source domains to address the scarcity problem. We also propose CD-LAPL to capture the transition patterns across different domains and generate the users' cross-domain local attentive preferences. These modules are all based on attention mechanism that can accelerate the training by parallel computation. Moreover, it can be applied to scenarios with more than one source domains, which is also demonstrated in the experiments.

Algorithm 1 describes the training procedure of our TJAPL. First, we calculate the target-domain attentive preference $\boldsymbol{f}_t$ through TD-APL (line 5), which is fed with the sequence of target domain. Next, we calculate the cross-domain user attentive preference $\boldsymbol{f}_t^u$ through CD-UAPL (line 6) and the cross-domain local attentive preference $\boldsymbol{f}_t^{S_n}$ via CD-LAPL (lines 7 - 9), which take the target-domain sequence and the source-domain sequences as input. Then, we aggregate all features to obtain the final representation of user's preference $\boldsymbol{o}_t$ (line 10). Finally, we calculate the prediction score $r_{t,i}$ for item i at time step t (line 11). We optimize our proposed model by minimizing the loss function $\mathcal{L}$ (lines 12 - 13).

The main technical contributions of this work can be briefly summarized as follows:

- We study a new and important problem, i.e., cross-domain sequential recommendation, and propose a novel method named transfer via joint attentive preference learning (TJAPL), which addresses the challenges well by transferring knowledge from more than one source domains to a target domain.

- We design cross-domain user attentive preference learning (CD-UAPL) to deal with the data scarcity problem by leveraging the users' overall preferences from the source domains, and design cross-domain local APL (CD-LAPL) to extract the transition patterns across different domains.

- We conduct extensive empirical studies on three cross-domain datasets, where the results show that our TJAPL significantly outperforms in all cases. We also conduct ablation studies to explore the contribution of various components of our TJAPL.

Table 3: Statistic details of datasets.

Dataset	# Overlapped-Users	# Items	# Interactions	Avg. Length	Density
Movie	10929	60902	462314	42.30	0.07%
CD		94171	348746	31.91	0.03%
Book		242363	615912	56.36	0.02%

5 Results and analysis

In this section, we introduce the experimental settings and conduct extensive empirical studies to answer the following six research questions:

(RQ1) What's the performance of our proposed TJAPL as compared with the state-of-the-art methods?

(RQ2) What's the influence of various components in our TJAPL?

(RQ3) How does the key parameters affect the performance of our TJAPL?

5.1 Datasets

We conduct empirical studies on Amazon[2], which is a review data collected by[35] from the eponymous e-commerce website. The Amazon data is suitable for the study of cross-domain sequential recommendation compared with other recommendation data, since it contains overlapped users in multiple domains. We choose three datasets with different categories, i.e., "Movie", "CD" and "Book". Then we follow[3,26] and preprocess these three datasets as follows: 1) We assume that all user interactions with items are positive feedback and determine the order of interactions by their timestamps. 2) We only keep the users and items with no fewer than five related interactions. And we drop duplicated (user, item) pairs. 3) We only keep the sequence of a user who has interactions in all the three domains. 4) We use the leave-one-out method for evaluation, which splits the sequence of each user into three parts, i.e., the last interaction for test, the penultimate interaction for validation and the remaining interactions for training. The statistics of the processed datasets is shown in Table 3.

5.2 Evaluation Metrics

We apply two common ranking-based metrics for the evaluation of recommendation performance, i.e., HT@10 (hit ratio) and NDCG@10 (normalized discounted cumulative gain), where the former is equivalent to recall because each user has exactly one preferred item in the test data in our case. Specifically, HT@10 denotes to the proportion of ground-truth items presenting in the top-10 recommended lists, while NDCG@10 is sensitive to the exact ranking positions of the items in recommended lists. We follow the common strategy in[3,12] and sample 100 negative items as candidates to avoid heavy computation on all (user, item) pairs. These 100 negative items have not been interacted with by the users and are sampled according to their popularity to ensure that they are informative and representative[26].

[2]http://jmcauley.ucsd.edu/data/amazon/

5.3 Baselines

To justify the effectiveness of our TJAPL, we compare it with ten recent and competitive methods from four recommendation categories. We adopt one general recommendation method (i.e., BPRMF) and one cross-domain general recommendation method (i.e., CoNet) since these tradition models typically do not perform well in the sequential recommendation task. We adopt five sequential recommendation methods (i.e., FPMC, GRU4Rec, GRU4Rec+, Caser and SASRec), including MCs-based model, RNN-based model, CNN-based model and attention-based model. For cross-domain sequential recommendation, we adopt three cross-domain sequential methods, including an RNN-based model π-net, a GNN-based model DA-GCN and an attention-based model CD-SASRec. These methods are all recently proposed and representative ones for the studied problem..

- BPRMF[9]. A classic model for general recommendation which optimize the matrix factorization by a pairwise ranking loss.
- CoNet[17]. A neural transfer learning model for general cross-domain recommendation through a collaborative cross-network[36]. It adds cross-connection units on MLP to enable dual information transfer.
- FPMC[18]. A traditional method for sequential recommendation that combines matrix factorization (MF) and first-order MCs. This method mainly models sequential information through MCs.
- GRU4Rec[1]. An RNN-based method for sequential recommendation that employs GRU to model users' behavior sequences step by step.
- GRU4Rec+[2]. An improved model based on GRU4Rec[1] that develops a new loss function and an additional sampling strategy.
- Caser[20]. A CNN-based model for sequential recommendation that adopts convolutional filters to the embeddings of the most recent items in order to capture high-order Markov chains.
- SASRec[3]. An attention based model that explores the sequential dependencies by adopting the attention mechanism. It also works as the target-domain attentive preference learning module in our TJAPL.
- π-Net[6]. An RNN-based model for cross-domain sequential recommendation in the shared-account scenario. It adopts a cross-domain transfer unit to capture and transfer user information across domains.
- DA-GCN[33]. A novel GNN-based model which links different domains by constructing a cross-domain sequence graph. It employs GNN to model the complicated interaction relationships, as well as the explicit structural information.
- CD-SASRec[30]. An improved method based on SASRec[3] for cross-domain sequential recommendation. It fuses the source-domain aggregated vector into the target-domain item embedding to transfer information across domains.

5.4 Implementation Details

We implement GRU4Rec[3], Caser[4], SASRec[5] and π-net [6] following the published codes of the original papers. The latent dimensionality d is selected from $\{10, 20, 30, 40, 50\}$ and chosen $d = 50$ since we find that on such sparse datasets, these methods usually benefit from a larger value of d[3,20]. For our TJAPL, we use Adam optimizer with a learning rate of 0.001, and the mini-batch size is set to 128, the dropout rate is set to 0.5. For all datasets, we set the maximum length of a sequence L to 100. The negative sampling number is set to 2048 for GRU4Rec+, the vertical and horizontal filter numbers are set to 4 and 16, respectively, for Caser, and other key parameters are followed the suggestions of the corresponding papers or turned on the validation data. For the architecture of attention-based methods (i.e., SASRec, CD-SASRec and our TJAPL), we adopt single-head attention layers and two attention blocks (i.e., $B = 2$). For the shared-account recommendation methods (i.e., π-Net and DA-GCN), the latent user number is set to 1.

For cross-domain recommendation methods, we only report the best performance of models with the corresponding source domain (i.e., when the target domain is Movie, we use CD or Book as a source domain to assist in training, and show only the best results).

5.5 Overall Performance Comparison (RQ1)

Table 4 illustrates the experimental results of our TJAPL and baselines on three datasets. We mark the best result in each column in bold and the second best result in underline..

Firstly, we can observe that our proposed TJAPL outperforms all the baselines on all the three datasets, and gains 14.89% NDCG@10 and 11.22% HR@10 improvements on average against the strongest baseline, which demonstrates the capability of our TJAPL to model the sequential information with cross-domain data. Besides, the sequential recommendation methods outperform the general recommendation baseline, which indicates the importance of extracting sequential information from users' behavior. And the cross-domain sequential recommendation methods outperform most traditional sequential recommendation methods, which demonstrates the significance of taking into account the cross-domain information. Moreover, the attention-based models achieve outstanding performances in both sequential recommendation and cross-domain sequential recommendation, which demonstrates the superiority of the attention mechanisms in modeling dynamic preference. Furthermore, among the three datasets, the "Movie" dataset has the most significant improvement, which probably because the "Movie" dataset is more tightly related to the other domains, i.e., a user's interaction sequences in the "Book" and "CD" domains are likely to influence his/her next interaction in the "Movie" domain, so knowledge transfer is more effective. Additionally, the cross-domain sequential recommendation methods achieve relatively small improvements on the "Book" dataset, since the source domain ("Movie" or "CD") is sparser (as is shown in Table 3). And our TJAPL can still achieve superior performance on the "Book" dataset because it can utilize both the "Movie" domain and the "CD" domain as source domains simultaneously, which

[3]https://github.com/hidasib/GRU4Rec

[4]https://github.com/graytowne/caser_pytorch

[5]https://github.com/kang205/SASRec

[6]https://github.com/mamuyang/PINet

Table 4: Recommendation performance of one general recommendation method (i.e., BPR), one general cross-domain recommendation method (i.e., CoNet), five sequential recommendation methods (i.e., FPMC, GRU4Rec, GRU4Rec+, Caser, SASRec), three cross-domain sequential recommendation method (i.e., π-Net, DA-GCN, CD-SASRec) and our TJAPL leveraging two source domains, on three datasets. Notice that for CoNet, π-net, DA-GCN and CD-SASRec, we report the better results when transferring knowledge from one of the other two source domains.

Method	Movie		CD		Book	
	NDCG@10	HT@10	NDCG@10	HT@10	NDCG@10	HT@10
BPRMF	0.0597	0.1256	0.0492	0.1142	0.0465	0.1088
CoNet	0.0675	0.1489	0.0756	0.1484	0.0764	0.1819
FPMC	0.0723	0.1697	0.0819	0.1785	0.0695	0.1416
GRU4Rec	0.1017	0.1984	0.1210	0.2247	0.1066	0.2162
GRU4Rec+	0.1133	0.2157	0.1440	0.2536	0.1293	0.2407
Caser	0.1231	0.2243	0.1267	0.2473	0.1163	0.2274
SASRec	0.1822	0.3234	0.1978	0.3569	0.1401	0.2607
π-Net	0.1113	0.2080	0.1265	0.2335	0.1042	0.2101
DA-GCN	0.1736	0.3124	0.1897	0.3458	0.1283	0.2375
CD-SASRec	0.1789	0.3173	0.2009	0.3614	0.1481	0.2737
TJAPL	**0.2133**	**0.3769**	**0.2199**	**0.3907**	**0.1632**	**0.2984**

demonstrates the effectiveness of knowledge transfer across multiple domains.

5.6 Ablation Study (RQ2)

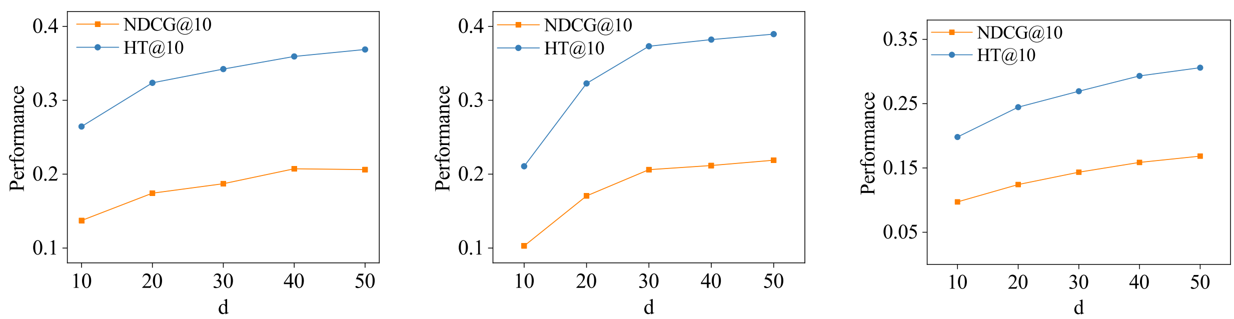
We conduct an ablation study to evaluate the contribution of different components of our TJAPL, and the results are presented in Table 5. In particular, we only report the results of “Movie” and “CD” when leveraging the “Book” domain for knowledge transfer, and leveraging the “Movie” domain for “Book”. We compare the separate effect of TD-APL (i.e., SASRec, denoted as ‘T’) with the joint effects that additionally add CD-UAPL (denoted as ‘U’) and CD-LAPL (denoted as ‘C’). We also examine the effects of different domains on CD-UAPL, i.e., target-domain user attentive preference learning (denoted as ‘U1’) and source-domain user attentive preference learning (denoted as ‘U2’). Moreover, we compare the joint effects of all the combination approaches.

Our observations are as follows.

- **‘T + U’ vs. T.** The integrated model with the addition of CD-UAPL always significantly outperforms the separate one, which demonstrates the importance of capturing the cross-domain user attentive preference and indicates the effectiveness of our CD-UAPL.
- **‘T + U’ vs. ‘T + U1’ or ‘T + U2’.** CD-UAPL is considered as the combination of the target-domain and source-domain user attentive preference learning modules. We can find that ‘T + U1’ is generally more effective than ‘T + U2’ (except on “Movie”) which means that users tend to generate the corresponding user preferences by applying their own target-domain data when it is sufficient. Furthermore, ‘T + U’ achieves the best overall performance, which indicates the benefit of combining the target-domain and

Table 5: Recommendation performance in ablation studies of our TJAPL with different architectures. Notice that 'T', 'U', 'C', 'U1', 'U2' represent TD-APL, CD-UAPL, CD-LAPL, target-domain UAPL and source-domain UAPL, respectively.

Architecture \ Setting	Book → Movie		Book → CD		Movie → Book	
	NDCG@10	HT@10	NDCG@10	HT@10	NDCG@10	HT@10
T	0.1840	0.3291	0.1978	0.3569	0.1401	0.2607
T + U	0.1864	0.3531	**0.2201**	<u>0.3881</u>	<u>0.1665</u>	**0.3076**
T + U1	0.1822	0.3445	0.2154	0.3842	0.1629	0.2953
T + U2	0.1876	0.3459	0.2118	0.3775	0.1619	0.2924
T + C	<u>0.1922</u>	<u>0.3586</u>	0.2097	0.3756	0.1579	0.2883
T + C + U	**0.2062**	**0.3687**	<u>0.2189</u>	**0.3895**	**0.1684**	<u>0.3059</u>



(a) Movie (b) CD (c) Book

Figure 3: Performance of different dimensionalities d on three datasets ($B = 2$).

source-domain user attentive preference.

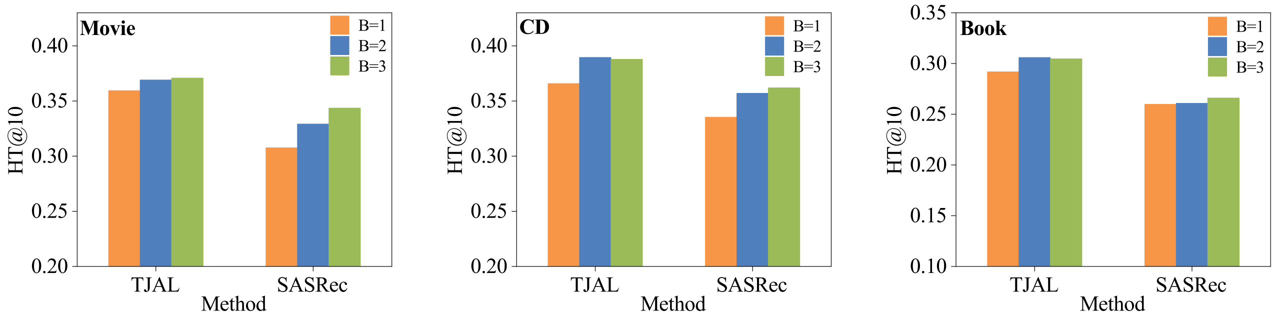
- **'T + C' vs. T.** Without CD-LAPL (i.e., 'C'), we find that the performance is much worse. It confirms that this module can learn the cross-domain local attentive preference from the recent interactions of the target and source domain, which indicates the significance of capturing the transition patterns across sequences from different domains.

- **'T + C + U' vs. 'T + U' or 'T + C'.** We can see that almost all the best results are from 'T + C + U', which demonstrates the complementarity of these three parts. It captures the local attentive preference and user attentive preference from both the target and source domains, balancing these representations and improving the effect for sequential recommendation.

5.7 Influence of Hyper-parameters (RQ3)

In this subsection, we explore the influence of two hyper-parameters (i.e., the latent dimensionality d and the number of attention blocks B) on the model performance. The results are presented in Fig. 3 and Fig. 4, respectively.

From Fig. 3, we observe that our model typically benefits from some relatively larger values of the dimensionality d, and it tends to be stable with $d \geqslant 40$ on all datasets. This means that a larger dimensionality does not always result in the better performance due to the overfitting problem.



(a) Movie (b) CD (c) Book

Figure 4: Performance (HT@10) of different numbers of blocks B ($d = 50$) for SASRec and our TJAPL.

From Fig. 4, we observe that unlike SASRec, it is sufficient to get the best performance for our TJAPL in most cases by setting the number of attention blocks $B = 2$, and stacking more blocks may not further improve the performance. That' s because in the hierarchical structure, the feature learned by SASRec in the bottom attention block can be seen as the long-term preference, which is similar to the user attentive preference learned in our TJAPL, and the increased model capacity may lead to overfitting.

6 Conclusion and future work

In this work, we propose an effective transfer learning solution called transfer via joint attentive preference learning (TJAPL) to deal with a new and important problem, i.e., cross-domain sequential recommendation. Specifically, we tackle the studied problem via attentive preference learning (ALP), including target-domain APL (TD-APL), cross-domain user APL (CD-UAPL) and cross-domain local APL (CD-LAPL). We treat the SASRec model as TD-APL to capture the dynamic preference in the target domain. Furthermore, we design CD-UAPL to share and transfer the user's overall preference from more than one source domains to the target domain, leveraging the behavior sequences from the source domains to address the data scarcity problem. We also design CD-LAPL to explore the transition patterns across sequences from different domains and capture the user's dynamic interests at each time step reflected from different domains. Extensive empirical studies on three real cross-domain datasets demonstrate that our TJAPL with knowledge transferred from one or two source domains outperforms the competitive baselines in all cases.

In the future, we aim to improve our model in a multi-target cross-domain recommendation scenario, which suffers from a more serious negative transfer problem since the knowledge in each domain is transferred to other domains more than once. Moreover, we are interested in studying our TJAPL in scenes of cross-domain or cross-organization privacy-aware federated recommendation[37], which can reduce the risk of privacy leakage from the introduction of rich source-domain data.

References

[1] HIDASI B, KARATZOGLOU A, BALTRUNAS L, et al. Session-based Recommendations with Re-

current Neural Networks[C]//ICLR '16: Proceedings of the 4th International Conference on Learning Representations. 2016.

[2] HIDASI B, KARATZOGLOU A. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations[C]//CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 843-852.

[3] KANG W, MCAULEY J J. Self-Attentive Sequential Recommendation[C]//ICDM '18: Proceedings of the 18th IEEE International Conference on Data Mining. 2018: 197-206.

[4] SUN F, LIU J, WU J, et al. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer[C]//CIKM '19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 1441-1450.

[5] ZHU F, WANG Y, CHEN C, et al. Cross-Domain Recommendation: Challenges, Progress, and Prospects [C]//IJCAI '21: Proceedings of the 30th International Joint Conference on Artificial Intelligence. 2021: 4721-4728.

[6] MA M, REN P, LIN Y, et al. π-net: A Parallel Information-Sharing Network for Shared-Account Cross-Domain Sequential Recommendations[C]//SIGIR '19: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 685-694.

[7] SUN W, MA M, REN P, et al. Parallel Split-Join Networks for Shared Account Cross-domain Sequential Recommendations[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.

[8] PATEREK A. Improving Regularized Singular Value Decomposition for Collaborative Filtering[C]// Proceedings of KDD Cup and Workshop. 2007: 39-42.

[9] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian Personalized Ranking from Implicit Feedback[C]//UAI '09: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. 2009: 452-461.

[10] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based Collaborative Filtering Recommendation Algorithms[C]//WWW '01: Proceedings of the 10th International Conference on World Wide Web. 2001: 285-295.

[11] AIOLLI F. Efficient Top-N Recommendation for Very Large Scale Binary Rated Datasets[C]//RecSys '13: Proceedings of the 7th ACM Conference on Recommender Systems. 2013: 273-280.

[12] HE X, LIAO L, ZHANG H, et al. Neural Collaborative Filtering[C]//WWW '17: Proceedings of the 26th International Conference on World Wide Web. 2017: 173-182.

[13] LI X, SHE J. Collaborative Variational Autoencoder for Recommender Systems[C]//KDD '17: Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2017: 305-314.

[14] MAN T, SHEN H, JIN X, et al. Cross-Domain Recommendation: An Embedding and Mapping Approach[C]//IJCAI '17: Proceedings of the 26th International Joint Conference on Artificial Intelligence: vol. 17. 2017: 2464-2470.

[15] LI P, TUZHILIN A. DDTCDR: Deep Dual Transfer Cross Domain Recommendation[C]//WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining. 2020: 331-339.

[16] SINGH A P, GORDON G J. Relational Learning via Collective Matrix Factorization[C]//KDD '08: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2008: 650-658.

[17] HU G, ZHANG Y, YANG Q. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation[C]//CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 667-676.

[18] RENDLE S, FREUDENTHALER C, SCHMIDT-THIEME L. Factorizing Personalized Markov Chains for Next-basket Recommendation[C]//WWW '10: Proceedings of the 19th International Conference on World Wide Web. 2010: 811-820.

[19] HE R, MCAULEY J. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation[C]//ICDM '16: Proceedings of the 16th IEEE International Conference on Data Mining. 2016: 191-200.

[20] TANG J, WANG K. Personalized top-N Sequential Recommendation via Convolutional Sequence Embedding[C]//WSDM '18: Proceedings of the 11th ACM International Conference on Web Search and Data Mining. 2018: 565-573.

[21] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All You Need[C]//NeurIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.

[22] YING H, ZHUANG F, ZHANG F, et al. Sequential Recommender System Based on Hierarchical Attention Network[C]//IJCAI '18: Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2018: 3926-3932.

[23] WU S, TANG Y, ZHU Y, et al. Session-based Recommendation with Graph Neural Networks[C]//AAAI '19: Proceedings of the 33rd AAAI Conference on Artificial Intelligence. 2019: 346-353.

[24] XU C, ZHAO P, LIU Y, et al. Graph Contextualized Self-attention Network for Session-based Recommendation[C]//IJCAI '19: Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019: 3940-3946.

[25] HE Y, ZHANG Y, LIU W, et al. Consistency-Aware Recommendation for User-Generated Item List Continuation[C]//WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining. 2020: 250-258.

[26] LIN J, PAN W, MING Z. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential recommendation[C]//RecSys '20: Proceedings of the 14th ACM Conference on Recommender Systems. 2020: 130-139.

[27] MA M, REN P, CHEN Z, et al. Mixed Information Flow for Cross-domain Sequential Recommendations [J]. ACM Transactions on Knowledge Discovery from Data, 2022, 16(4): 1-32.

[28] OUYANG W, ZHANG X, ZHAO L, et al. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction[C]//CIKM '20: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. 2020: 2669-2676.

[29] LI P, JIANG Z, QUE M, et al. Dual Attentive Sequential Learning for Cross-Domain Click-Through Rate Prediction[C]//KDD '21: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2021: 3172-3180.

[30] ALHARBI N, CARAGEA D. Cross-Domain Self-Attentive Sequential Recommendations[C]//ICONDATA '22: Proceedings of International Conference on Data Science and Applications. 2022: 601-614.

[31] ZHANG T, ZHAO P, LIU Y, et al. Feature-level Deeper Self-Attention Network for Sequential Recommendation[C]//IJCAI '19: Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019: 4320-4326.

[32] KABBUR S, NING X, KARYPIS G. FISM: Factored Item Similarity Models for top-N Recommender Systems[C]//KDD '13: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2013: 659-667.

[33] GUO L, TANG L, CHEN T, et al. DA-GCN: A Domain-aware Attentive Graph Convolution Network for Shared-account Cross-domain Sequential Recommendation[C]//IJCAI '21: Proceedings of the 13th International Joint Conference on Artificial Intelligence. 2021: 2483-2489.

[34] KINGMA D P, BA J. Adam: A Method for Stochastic Optimization[C]//ICLR '15: Proceedings of the 3rd International Conference on Learning Representations. 2015.

[35] MCAULEY J, TARGETT C, SHI Q, et al. Image-based Recommendations on Styles and Substitutes[C]//SIGIR '15: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2015: 43-52.

[36] MISRA I, SHRIVASTAVA A, GUPTA A, et al. Cross-Stitch Networks for Multi-task Learning[C]//CVPR '16: 2016 IEEE Conference on Computer Vision and Pattern Recognition. 2016: 3994-4003.

[37] LIN Z, PAN W, YANG Q, et al. Recommendation Framework via Fake Marks and Secret Sharing[J]. ACM Transactions on Information Systems, 2022.