Neural Attentive Cross-Domain Recommendation

Dimitrios Rafailidis
Maastricht University
Maastricht, The Netherlands
dimitrios.rafailidis@maastrichtuniversity.nl

Fabio Crestani
Faculty of Informatics
Università della Svizzera italiana (USI)
Lugano,Switzerland
fabio.crestani@usi.ch

ABSTRACT

Nowadays, users open multiple accounts on social media platforms and e-commerce sites, expressing their personal preferences on different domains. However, users' behaviors change across domains, depending on the content that users interact with, such as movies, music, clothing and retail products. The main challenge is how to capture users' complex preferences when generating cross-domain recommendations, that is exploiting users' preferences from source domains to generate recommendations in a target domain. In this study, we propose a Neural Attentive Cross-domain model, namely NAC. We design a neural architecture, to carefully transfer the knowledge of user preferences across domains by taking into account the cross-domain latent effects of multiple source domains on users' selections in a target domain. In addition, we introduce a cross-domain behavioral attention mechanism to adaptively perform the weighting of users' preferences from the source domains, and consequently generate accurate cross-domain recommendations. Our experiments on ten cross-domain recommendation tasks show that the proposed NAC model achieves higher recommendation accuracy than other state-of-the-art methods for both ordinary and cold-start users. Furthermore, we study the effect of the proposed cross-domain behavioral attention mechanism and show that it is a key factor to our model's performance.

CCS CONCEPTS

• **Information systems → Collaborative and social computing systems and tools**.

KEYWORDS

Recommendation systems; cross-domain recommendation; neural attentive models

ACM Reference Format:
Dimitrios Rafailidis and Fabio Crestani. 2019. Neural Attentive Cross-Domain Recommendation. In *The 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19), October 2–5, 2019, Santa Clara, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341981.3344214

ICTIR '19, October 2–5, 2019, Santa Clara, CA, USA

ACM ISBN 978-1-4503-6881-0/19/10…$15.00
https://doi.org/10.1145/3341981.3344214

1 INTRODUCTION

The collaborative filtering strategy, where users with similar preferences tend to get similar recommendations, has been widely followed in recommendation systems. User preferences are expressed explicitly in the form of ratings or implicitly in the form of number of views, clicks, purchases, and so on. Representative collaborative filtering strategies are matrix factorization techniques, which factorize the data matrix with user preferences in a single domain (e.g., music or video), to reveal the latent associations between users and items. However, data sparsity and cold-start problems degrade the recommendation accuracy, as there are only a few preferences on which to base the recommendations in a single domain [14]. With the advent of social media platforms and e-commerce systems, such as Amazon and Netflix, users express their preferences in multiple domains. For example, in Amazon users can rate items from different domains, such as books and retail products, and users express their opinion on different social media platforms, such as Facebook and Twitter. In the effort to overcome the data sparsity and cold-start problems, several cross-domain recommendation strategies have been proposed, which exploit the additional information of user preferences in multiple auxiliary/source domains to leverage the recommendation accuracy in a target domain [6]. However, generating cross-domain recommendations is a challenging task [5]. For example, if the source domains are richer than the target domain, algorithms learn how to recommend items in the source domains and consider the target domain as noise. Moreover, the source domains might be a potential source of noise, for example, if user preferences differ in the multiple domains, the source domains introduce noise in the learning of the target domain. Therefore, a pressing challenge resides on how to transfer the knowledge of user preferences from different domains by also weighting the importance of users' different behaviors accordingly.

In cross-domain recommendation, the source domains can be categorized based on users' and items' overlaps, that is, full-overlap, and partial or non user/item overlap between the domains [5]. In this study, we focus on partial users' overlaps between the target and the source domains, as it reflects on the real-world setting [6]. Relevant methods, such as [8, 13, 15, 17], form user and item clusters to capture the relationships between multiple domains at a cluster level, thus tackling the sparsity problem; and then weigh the user preferences to generate the top-N recommendations in the target domain. However, existing cross-domain strategies linearly combine the cluster-based user preferences in the target domain, which does not reflect on the real-world world scenario with users having complex behaviors across domains.

The key factors to generate accurate cross-domain recommendation are the capturing of users' different preferences and the

weighting of the importance of users' behaviors, accordingly. Recently, attention mechanisms have been shown to be effective in various tasks such as image captioning [30] and machine translation [3], among others. Essentially the idea behind such mechanisms is that the outputs of neural models depend on *'relevant'* parts of some input that the models should focus on. Armed with different attention mechanisms, single-domain recommendation models have been designed to generate sequential [16, 18], social [27], and context-aware recommendations [28, 29]. Nonetheless, these attention models produce recommendations in a single-domain, and omit users' various and complex behaviors across domains.

To overcome the shortcomings of existing cross-domain recommendation strategies, we propose a Neural Attentive Recommendation model, namely NAC, making the following contributions:

- We design a neural architecture to carefully transfer the knowledge of user preferences across domains, by taking into account the cross-domain latent effects of multiple source domains on users' selections in a target domain.
- We propose a cross-domain behavioral attention mechanism to adaptively perform the weighting of users preferences from the source domains, and therefore focus on a subset of users that have similar behaviors across the domains.

In our experiments on ten cross-domain tasks we demonstrate the effectiveness of the proposed NAC model compared to other state-of-the-art methods. The remainder of the paper is organized as follows, Section 2 reviews related work, and then Section 3 details the proposed NAC model. Finally, in Section 4 we examine the performance of the proposed model against both single-domain and cross-domain baselines on the ten cross-domain tasks, and Section 5 concludes the study.

2 RELATED WORK

2.1 Cross-domain Recommendation

Cross-domain recommendation algorithms differ in how the knowledge of user preferences from the source domains is exploited, when generating the recommendations in the target domain. Various cross-domain approaches aggregate user preferences into a unified matrix, on which weighted single-domain techniques are applied, such as user-based kNN [4]. The graph-based method presented in [6] models the similarity relationships as a direct graph and explores all possible paths connecting users or items to capture the cross-domain relationships. Pan et al. [21] transform the knowledge of user preferences from different domains with heterogenous forms of user explicit or implicit feedback, to compute the shared latent features. Hu et al. [13] model a cubic user-item-domain matrix (tensor), and by applying factorization the respective latent space is constructed, based on which the cross-domain recommendations are generated. Li et al. [15] calculate user and item clusters for each domain, and then encode the cluster-based patterns in a shared codebook. Finally, the knowledge of user preferences is transferred across domains through the shared codebook. Gao et al. [8] compute the latent factors of user-clusters and item-clusters to construct a common latent space, which represents the preference patterns e.g., rating patterns, of user clusters on the item clusters. Then, the common cluster-based preference pattern that is shared across domains is learned following a subspace strategy, so as to control the optimal level of sharing among multiple domains. Cross-Domain collaborative filtering with factorization machines (FM), presented in [17], is a state-of-the-art cross-domain recommendation which extends FM [25]. It is a context-aware approach which applies factorization on the merged domains, aligned by the shared users, where the source domains are used as context. Hu et al. [12] jointly learn neural networks to generate cross-domain recommendations based on stich units [20], introducing a shared auxiliary matrix to couple two hidden layers when training the networks in parallel. However, these cross-domain recommendation strategies do not pay attention to users' complex behaviors across domains, where only a subset of users' preferences match while transferring the knowledge of users' selections from multiple domains.

2.2 Neural Attention Models

More recently, neural attention models have been introduced for single-domain recommendation tasks. For example, Ebesu et al. [7] propose collaborative memory networks, where the associative addressing scheme of the memory module acts as a nearest neighborhood model identifying similar users. The attention mechanism learns an adaptive nonlinear weighting of the user's neighborhood based on the specific user and item. The output module exploits nonlinear interactions between the adaptive neighborhood state jointly with the user and item memories to derive the recommendation. Tay et al. [28] present a neural attention model to weigh the gravity of users' reviews on items, assuming that not all reviews are created equal, but only a selected few are important. Liu et al. [16] incorporate attention weights in recurrent neural networks as a priority model to distinguish current interests e.g., clicks from long term preferences. Manotumruksa et al. [18] introduce a contextual attention gate that controls the influence of the ordinary context on the users' contextual preferences and a time- and geo-based gate that controls the influence of the hidden state from the previous check-in based on the transition context. Sun et al. [27] investigate the problem of how to leverage social influence to enhance the temporal social recommendation performance, introducing an attentive recurrent network based approach. Attentional factorization machines learn the importance of each feature interaction for content-aware recommendation [29]. Nonetheless all the aforementioned attention-based models generate recommendations for a single-domain, and do not account for users' different preferences across several domains. To the best of our knowledge our work is the first study that investigates a behavioral attention mechanism for cross-domain recommendation.

3 THE PROPOSED MODEL

Our notation is presented in Table 1. We assume that we have d different domains, where n_p and m_p are the numbers of users and items in the p-th domain, respectively. In the matrix $\mathbf{R}_p$, we store the user preferences on items, in the form of explicit feedback e.g., ratings or in the form of implicit feedback e.g., number of views, clicks, and so on. In this study we consider users' partial overlaps across the domains. We define a users' overlapping matrix $\mathbf{X}_{pt} \in \mathbb{R}^{n_p \times n_t}$ between a source domain p and a target domain t. For each cell, holds $(\mathbf{X}_{pt})_{ab} = 1$, if users a and b are the same user in the source and target domains, and 0 otherwise. The goal of the

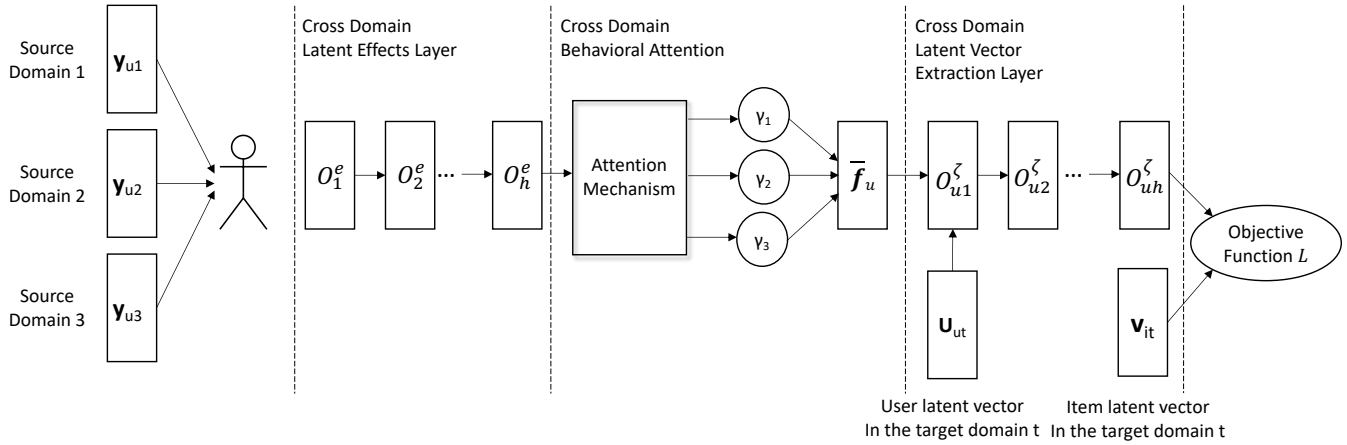

Figure 1: An overview of the proposed NAC model. Our architecture consists of the *cross-domain latent effects layer*, the *cross-domain behavioral attention mechanism* and the *cross-domain latent vector extraction layer*. The model parameters are learned based on the objective function $\mathcal{L}$ via backpropagation.

Table 1: Notation.

Symbol	Description
d	Number of domains
n_p	Number of users in the p-th source domain, $p = 1, \ldots, d-1$
m_p	Number of items in the p-th domain
l	Number of latent dimensions
h	Number of hidden layers
$\mathbf{R}_p \in \mathbb{R}^{n_p \times m_p}$	User-item interaction (rating) matrix in the p-th domain
$\mathbf{A}_p \in \mathbb{R}^{n_p \times n_p}$	Adjacency matrix of the users' graph in the p-th domain
c_p	Number of user clusters in the p-th domain
$\mathbf{C}_p \in \mathbb{R}^{n_p \times c_p}$	Cluster assignment matrix in the p-th domain
$\mathbf{X}_{pt} \in \mathbb{R}^{n_p \times n_t}$	Users' overlapping matrix between p and target domain t
$\mathbf{Y}_{pt} \in \mathbb{R}^{c_p \times c_t}$	Cluster-based cross domain matrix between p and t
$\mathbf{u}_{up} \in \mathbb{R}^{l \times 1}$	Latent vector of user u in the p-th domain, $u = 1, \ldots, n_p$
$\mathbf{v}_{ip} \in \mathbb{R}^{l \times 1}$	Latent vector of item i in the p-th domain
$\mathbf{y}_{up} \in \mathbb{R}^{l \times 1}$	Cluster-based cross-domain latent vector of user u in the p-th domain
$\mathbf{f}_{up} \in \mathbb{R}^{l \times 1}$	User's u latent vector of cross-domain effect in the p-th domain
$\mathbf{z}_{up} \in \mathbb{R}^{l \times 1}$	Cross-domain latent vector of user u
$\mathbf{W}_q^e \in \mathbb{R}^{l \times l}$	Weight matrix in the q-th hidden layer of model e, $q = 1, \ldots, h$
$\mathbf{o}_q^e \in \mathbb{R}^{l \times 1}$	Users' hidden representation in the q-th hidden layer of model e

proposed NAC model is to generate personalized recommendations in the target domain t, while transferring and weighting users' different preferences/behaviors from the $d-1$ source domains.

3.1 NAC Overview

Figure 1 illustrates an overview of the proposed NAC model. In the example presented in Figure 1, we consider three source domains, and a target domain t. We first follow a co-clustering strategy to group users based on their preferences in each source domain p and target domain t, with $p = 1, \ldots, d-1$. Then, by factorizing the respective cluster assignment matrices we extract cluster-based cross-domain latent vectors $\mathbf{y}_{up} \in \mathbb{R}^{l \times 1}$ in the p-th domain, with $u = 1, \ldots, n_p$ and l being the number of latent factors. In the *cross-domain latent effects layer*, we compute the latent effects of users across the different domains based on a Multi-Layer Perceptron (MLP) network. Then, in the *cross-domain behavioral attention mechanism*, we calculate the attention weights γ_p of the latent effects, which correspond to how much users' preferences match across the source domains and the target domain. In the *cross-domain latent vector extraction layer* we capture the nonlinear associations between the aggregated cross-domain latent effects of the source domains, that is vector $\bar{\mathbf{f}}_u \in \mathbb{R}^{l \times 1}$, and the user latent vector $\mathbf{u}_{ut}$ in the target domain t via a MLP network. The output is a cross-domain user latent vector $\mathbf{o}_{uh}^{\zeta} = \mathbf{z}_{ut} \in \mathbb{R}^{l \times 1}$ which is combined with the item latent vector $\mathbf{v}_{it} \in \mathbb{R}^{l \times 1}$ to learn the objective function $\mathcal{L}$ of our model. At this point we would like to mention that although the architecture of Figure 1 is designed for users' partial overlaps across domains, it is easy to extend the proposed NAC model for item' partial overlaps by designing a respective neural architecture for both users and items in parallel, thus computing both users' and items' cross-domain latent vectors.

The remainder of the Section is structured as follows, Section 3.2 presents the *co-clustering strategy* to group users based on their multi-preferences in a source domain p and a target domain t. Section 3.3 presents the *cross-domain latent effects layer* of the source domains, and Section 3.4 details the *cross-domain behavioral attention mechanism*. Finally, Section 3.5 presents the *cross-domain latent vector extraction layer*, and in Section 3.6 we formulate the objective function of the proposed NAC model, while Section 3.7 provides the implementation details.

3.2 Cross-domain Co-Clustering

Before starting with the cross-domain clustering strategy, we first capture the users' similar preferences in each p-th domain based on the matrix $\mathbf{R}_p$. If users a and b have interacted with at least a common item i, then users a and b are connected. The connections/similarities are stored in an adjacency matrix $\mathbf{A}_p$, whose ab-th entries are calculated as follows:

$$(\mathbf{A}_p)_{ab} = \begin{cases} \frac{\sum_{i=1}^{m_p} (\mathbf{R}_p)_{ai}(\mathbf{R}_p)_{bi}}{\sqrt{\sum_{i=1}^{m_p} (\mathbf{R}_p)_{ai}^2}\sqrt{\sum_{i=1}^{m_p} (\mathbf{R}_p)_{bi}^2}} & \text{, if users } a \text{ and } b \text{ are connected} \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

with $a, b = 1, \ldots, n_p$.

Given the adjacency matrix $\mathbf{A}_p$, first we have to define the objective function for performing user clustering on the p-th domain, that is, to calculate the cluster assignment matrix $\mathbf{C}_p \in \mathbb{R}^{n_p \times c_p}$, which corresponds to the following minimization problem:

$\forall p = 1, \ldots, d-1$

$$\min_{\mathbf{C}_p} \sum_{ab} ||(\mathbf{C}_p)_{a*} - (\mathbf{C}_p)_{b*}||^2, \quad \text{with } a, b = 1, \ldots, n_p \quad \text{subject to } \mathbf{C}_p^T\mathbf{C}_p = \mathbf{I}, \mathbf{C}_p \geq 0 \tag{2}$$

with orthogonality constraints on the cluster matrix $\mathbf{C}_p$, and the user assignments to clusters being 0 or positive. According to the Laplacian method of [9], the minimization problem of Equation (2) is equivalent to:

$$\min_{\mathbf{C}_p} \sum_{ab} ||(\mathbf{C}_p)_{a*} - (\mathbf{C}_p)_{b*}||^2 = \min_{\mathbf{C}_p} Tr(\mathbf{C}_p^T\mathbf{L}_p\mathbf{C}_p) \quad \text{subject to } \mathbf{C}_p^T\mathbf{C}_p = \mathbf{I}, \mathbf{C}_p \geq 0 \tag{3}$$

where $Tr(\cdot)$ is the trace operator. Matrix $\mathbf{L}_p \in \mathbb{R}^{n_p \times n_p}$ is the Laplacian of the adjacency matrix $\mathbf{A}_p$, which is computed as follows: $\mathbf{L}_p = \mathbf{D}_p - \mathbf{A}_p$, where $\mathbf{D} \in \mathbb{R}^{n_p \times n_p}$ is a diagonal matrix, whose entries are calculated as $(\mathbf{D}_p)_{aa} = \sum_{ab} (\mathbf{A}_p)_{ab}$. Similarly, we define the respective objective function in Equation (3), for performing user clustering on the target domain t, denoted by matrix $\mathbf{C}_t \in \mathbb{R}^{n_t \times c_t}$.

To compute the cluster-based similarities of users between domains p and t, we follow a co-clustering strategy for each source domain p and a target domain t, trying to minimize the following objective function:

$$\min_{\mathbf{Y}_{pt}} ||\mathbf{X}_{pt} - \mathbf{C}_p\mathbf{Y}_{pt}\mathbf{C}_t^T||_F^2 + \lambda||\mathbf{Y}_{pt}||_{2,1} \quad \text{subject to } \mathbf{Y}_{pt}^T\mathbf{Y}_{pt} = \mathbf{I}, \mathbf{Y}_{pt} \geq 0 \tag{4}$$

with orthogonality constraints on the cluster-based cross-domain matrix $\mathbf{Y}_{pt} \in \mathbb{R}^{c_p \times c_t}$, whose elements are 0 or positive. The symbol $||\cdot||_{2,1}$ denotes the $L_{2,1}$ norm of a matrix which is calculated as follows:

$$||\mathbf{Y}_{pt}||_{2,1} = \sum_{a=1}^{n_p} \sqrt{\sum_{b=1}^{n_t} (\mathbf{Y}_{pt})_{ab}{}^2} = \sum_{a=1}^{n_p} ||(\mathbf{Y}_{pt})_{a*}||_2 \tag{5}$$

The $L_{2,1}$ regularization term in Equation (4) forces the solution of matrix $\mathbf{Y}_{pt}$ to be sparse, reflecting on the real-world scenario, where users' overlaps are usually sparse [5]. Parameter $\lambda > 0$ controls the respective $L_{2,1}$ regularization term in Equation (4). Notice that the solution of matrix $\mathbf{Y}_{pt}$, corresponds to soft (overlapping) co-clusters, which reflects on the real-case with users having multi-preferences across domains, thus belonging to more than one user clusters.

3.3 Cross-domain Latent Effects Layer of the Source Domains

The goal of the first layer of NAC is to compute the latent effects of users across the different domains. To achieve this, we design a MLP network to model the users' cross-domain latent effects. By factorizing the respective $d-1$ matrices $\mathbf{Y}_{pt}$, we extract the cluster-based cross-domain latent vector $\mathbf{y}_{up} \in \mathbb{R}^{l \times 1}$ in the p-th domain for each user u. Then, we map the cluster-based cross-domain latent vector $\mathbf{y}_{up}$ and the user latent vector $\mathbf{u}_u$ in the target domain t into a shared embedding layer as follows:

$$\mathbf{o}_0^e = g(\mathbf{W}_{u0}^e\mathbf{u}_u + \mathbf{W}_{p0}^e\mathbf{y}_{up} + \mathbf{b}_0^e) \tag{6}$$

with $p = 1, \ldots, d-1$

Matrices $\mathbf{W}_{u0}^e$ and $\mathbf{W}_{p0}^e \in \mathbb{R}^{l \times l}$ are the weight matrices for the latent vector of user u and cluster-based cross-domain latent vector in the p-th domain, respectively, and $\mathbf{b}_0^e \in \mathbb{R}^{l \times 1}$ is the bias vector. $g(x) = \max(0, x)$ is the Rectifier (ReLU) activation function which is non-saturated. The saturation problem occurs when neurons stop learning and their output is near to either 0 or 1, a problem that might be suffered by the sigmoid and tanh functions [31].

Next, we stack h hidden layers on the top of the embedding layer, where the representation of each hidden layer q is computed as follows:

$$\mathbf{o}_q^e = g(\mathbf{W}_q^e\mathbf{o}_{q-1}^e + \mathbf{b}_q^e) \tag{7}$$

with $q = 1, \ldots, h$

Having computed the representation $\mathbf{o}_h^e$ of the last hidden layer h, to capture the cross-domain effect of user u in a source domain p we calculate user's u latent vector of cross-domain effect of p as follows:

$$\mathbf{f}_{up} = \mathbf{W}_{pf}^e\mathbf{o}_h^e + \mathbf{b}_{pf}^e \tag{8}$$

where $\mathbf{W}_{pf}^e \in \mathbb{R}^{l \times l}$ and $\mathbf{b}_{pf}^e \in \mathbb{R}^{l \times 1}$ denote the weight matrix and bias vector of the final user latent vector of cross-domain effect of domain p, respectively.

3.4 Cross-domain Behavioral Attention

To measure the importance of the $d-1$ cross-domain effects $\mathbf{f}_{up}$, we propose a *cross-domain behavioral attention mechanism*. The attention mechanism learns an adaptive weighting function to focus on a subset of users that have similar behavior across the domains. Provided the $d-1$ cross-domain effect vectors and the user latent vector $\mathbf{u}_{ut}$ in the target domain t, we employ a single-layer perceptron to calculate the respective attention score of a source domain $p = 1, \ldots, d-1$ for the target domain t as follows:

$$\phi(\mathbf{u}_{ut}, \mathbf{f}_{up}) = g(\mathbf{W}_1^\psi\mathbf{u}_{ut} + \mathbf{W}_2^\psi\mathbf{f}_{up} + \mathbf{b}^\psi) \tag{9}$$

where $\mathbf{W}_1^\psi$ and $\mathbf{W}_2^\psi \in \mathbb{R}^{l \times l}$ are the weight matrices, and $\mathbf{b}_{pf}^e \in \mathbb{R}^{l \times 1}$ is the bias. The superscript ψ refers to the model for the cross-domain behavioral attention mechanism. Then, the final weights are computed by normalizing the respective $d-1$ attention scores with the softmax function, which reflects on the importance of user's u cross-domain effect of domain p, as follows:

$\forall p = 1, \ldots, d-1$

$$\gamma(\mathbf{u}_{ut}, \mathbf{f}_{up}) = \frac{\exp(\phi(\mathbf{u}_{ut}, \mathbf{f}_{up})}{\sum_{v=1}^{d-1} \exp(\phi(\mathbf{u}_{ut}, \mathbf{f}_{uv})} \quad (10)$$

Notice that the attention mechanism selectively weighs the users' similarity on preferences across the different domains based on the attention scores $\gamma(\mathbf{u}_{ut}, \mathbf{f}_{up})$. Having calculated the $d-1$ cross-domain attention scores of the respective domains, the latent vector of aggregated cross-domain effect on user's u behavior in the target domain t is computed as follows:

$$\bar{\mathbf{f}}_u = \sum_{v=1}^{d-1} \gamma(\mathbf{u}_{ut}, \mathbf{f}_{uv}) \mathbf{f}_{uv} \quad (11)$$

3.5 Cross-domain Latent Vector Extraction Layer

The goal of the *cross-domain latent vector extraction layer* is to forecast how the effects of the users' different behaviors in the $d-1$ source domains influence the user representation in the target domain t. Similar to the *cross-domain latent effects layer* of Section 3.3, we first map the aggregated cross domain effect $\bar{\mathbf{f}}_u$ of all $d-1$ domains and the user latent vector $\mathbf{u}_{ut}$ of user u in the target domain t into a shared embedding layer, thus constructing a hidden representation $\mathbf{o}_{u0}^{\zeta} \in \mathbb{R}^{l \times 1}$, as shown in Figure 1. Then, vector $\mathbf{o}_{u0}^{\zeta}$ is fed to a MLP network of h hidden layers, to capture the nonlinear associations of the complex cross-domain effects on user's behavior in the target domain t. The output of the MLP network is the *cross-domain latent vector* $\mathbf{z}_{ut} = \mathbf{o}_{uh}^{\zeta} \in \mathbb{R}^{l \times 1}$, that is the last hidden representation of the MLP network.

3.6 Objective Function

The proposed NAC model aims at the ranking performance of the recommendations in the target domain t. Having computed the cross-domain latent vector $\mathbf{z}_{ut}$ for each user u in the target domain t, with $u = 1, \ldots, n_t$, we consider the respective item latent vectors $\mathbf{v}_{it}$, with $i = 1, \ldots, m_t$. In particular, we define two disjoint sets, a set $\mathcal{I}_u^+$ of observed items that user u has already interacted with in the target domain t, and a set $\mathcal{I}_u^-$ of unobserved items. For each observed item $i^+ \in \mathcal{I}_u^+$, we randomly sample negative/unobserved items $i^- \in \mathcal{I}_u^-$, for each user u. According to the Bayesian Pairwise Ranking (BPR) criterion [26], we try to rank the observed items higher than the unobserved ones, having the following loss function:

$$\mathcal{L} = - \sum_{(u, i^+, i^-)} \log \sigma(\hat{\mathbf{R}}_{tui^+} - \hat{\mathbf{R}}_{tui^-}) \quad (12)$$

where $\sigma(x) = 1/\Big(1 + \exp(-x)\Big)$ is the logistic sigmoid function. $\hat{\mathbf{R}}_{tui^+} = \mathbf{z}_{ut}^{\top} \mathbf{v}_{i^+t}$ is the respective cell of the factorized user-item interaction matrix for a user u and his/her observed item i^+ in the target domain t, and is computed as the product of the cross-domain latent vector $\mathbf{z}_{ut}$ of user u of Section 3.5 and the item latent vector $\mathbf{v}_{i^+t}$ of item i^+. Similarly, we calculate the term $\hat{\mathbf{R}}_{tui^-} = \mathbf{z}_{ut}^{\top} \mathbf{v}_{i^-t}$ based on the cross-domain latent vector of user u and the respective item latent vector of item i^- of a negative sample.

3.7 Implementation Details

In our implementation we used Tensorflow[1]. We computed the model's parameters, that is the weight matrices of Sections 3.3-3.5 via backpropagation with stochastic gradient descent, trying to solve the minimization problem of the ranking loss function $\mathcal{L}$ in Equation (12). We employed mini-batch Adam, which adapts the learning rate for each parameter by performing smaller updates for frequent and larger updates for infrequent parameters. We set the batch size of mini-batch Adam to 512 with a learning rate of 1e-4. In each backpropagation iteration we performed negative sampling to randomly select a subset $\mathcal{I}_u^-$ of unobserved items as negative instances. In our implementation we used five negative samples for each positive/observed sample, as we found out that for larger numbers of negative samples the computational cost of the model learning did not pay off in terms of recommendation accuracy. We varied the number of latent dimensions l from 10 to 100 by a step of 10, using a grid selection strategy and we kept the latent dimensions fixed based on cross validation.

In addition, to account for the fact that the gradient-based optimization strategy might find a locally - optimal solution of the model's parameter set, we followed a pretraining strategy. We first trained our model with random initializations using only one hidden layer in the MLP networks, employed in our neural architecture. Then, we used the trained parameters as the initialization of our model and varied the number of hidden layers from 1 to 5 by a step of 1, where we concluded in the optimal number of hidden layers using cross-validation. The pretraining strategy is very important for our model. To verify this we tested our model without applying the pretraining strategy and we found that there is an average drop of -5.27% in the model's performance. This observation has been also confirmed by other relevant studies pointing out that the initialization of the model parameters plays a significant role for the model's convergence and performance [10].

Table 2: The ten cross-domain recommendation tasks

Domain	Users	Items	Ratings	Density (%)
Baby Care	5,422	3,165	21,340	0.124
Books	15,507	59,346	108,887	0.011
Destinations	9,290	3,615	31,418	0.093
Music	16,002	35,807	96,226	0.016
Online Stores/Services	28,643	5,518	54,734	0.034
Personal Care	6,214	10,786	28,945	0.043
Sport/Outdoor	6,750	9,597	19,181	0.029
Toys	9,040	18,681	51,152	0.030
Used Cars	17,041	4,174	28,598	0.040
Video/DVD	25,218	28,972	175,665	0.024

4 EXPERIMENTS

4.1 Cross-domain Tasks

Our experiments were performed on ten cross-domain tasks from the Rich Epinions Dataset (RED), with 131,228 users, 317,775 items and 1,127,673 ratings at a 5-star scale, having users' partial overlaps across the domains [19]. The items are grouped in categories/domains,

[1] https://www.tensorflow.org

Table 3: Effect on recall. Bold values denote the best scores, using the paired t-test (p <0.05). The last column expresses the relative improvement of the proposed NAC model, compared to the second best method.

Target domain	BPR	NeuMF	CLFM	SCoNet	NAC*	NAC	Improvement (%)
Baby Care	0.4202	0.4673	0.5188	0.5250	0.5081	**0.5521**	**5.15**
Books	0.1134	0.1258	0.1449	0.1726	0.1599	**0.1873**	**8.51**
Destinations	0.3575	0.4261	0.4491	0.4612	0.4505	**0.4904**	**6.31**
Music	0.1747	0.1811	0.2132	0.2358	0.2276	**0.2583**	**9.54**
Online Stores	0.3398	0.3842	0.4465	0.4701	0.4382	**0.4908**	**4.40**
Personal Care	0.1569	0.1741	0.1884	0.2281	0.2134	**0.2536**	**11.17**
Sport/Outdoor	0.1239	0.1439	0.1715	0.1892	0.1833	**0.2029**	**7.22**
Toys	0.2888	0.3260	0.3558	0.3705	0.3526	**0.3912**	**5.58**
Used Cars	0.1085	0.1173	0.1305	0.1561	0.1479	**0.1618**	**3.65**
Video/DVD	0.3944	0.43038	0.4656	0.4872	0.4803	**0.5105**	**4.78**

and we evaluate the performance of our model on the ten largest domains. The main characteristics of the evaluation data are presented in Table 2.

4.2 Evaluation Setup

In each out of the ten cross-domain recommendation tasks, the goal is to generate recommendations for a target domain, while the remaining nine domains are considered as source domains. We trained the examined models on the 50% of the target domain and used all the ratings of the source domains as training set. We used 10% of the ratings in the target domain as cross-validation set to tune the models' parameters and evaluate the examined models on the remaining test ratings. To remove user rating bias from our results, we considered an item as relevant if a user has rated it above her average ratings and irrelevant otherwise. We measured the quality of the top-k recommendations in terms of the ranking-based metrics recall and Normalized Discounted Cumulative Gain (NDCG@k). Recall is the ratio of the relevant items in the top-k ranked list over all the relevant items for each user. NDCG measures the ranking of the relevant items in the top-k list. For each user the Discounted Cumulative Gain (DCG) is defined as:

$$DCG@k = \sum_{j=1}^{k} \frac{2^{rel_j} - 1}{\log_2 j + 1}$$

where rel_j represents the relevance score of item j, that is binary in our case, i.e., relevant or irrelevant. NDCG is the ratio of DCG/iDCG, where iDCG is the ideal DCG value given the ratings in the test set. We fixed the number of recommendations to k=10. We repeated our experiments five times and averaged recall and NDCG over the five runs.

4.3 Compared Methods

We compare the proposed NAC model with the following baselines: **BPR** [26]: a *single-domain* Bayesian Personalized Ranking strategy that tries to rank the observed items higher than the unobserved ones in the target domain. **NeuMF** [11]: a baseline *single-domain* Neural Matrix factorization scheme that follows the collaborative filtering strategy. NeuMF also exploits users' preferences only in the target domain. **CLFM** [8]: a *cross-domain* Cluster-based Latent Factor Model which uses joint nonnegative tri-factorization to construct a latent space to represent the rating patterns of user clusters on the item clusters from each domain, and then generates the cross-domain recommendations based on a subspace learning strategy. **SCoNet** [12]: a *cross-domain* model that jointly learns stich networks, with a shared auxiliary matrix to couple two hidden layers when training the networks in parallel. In our experiments, we used the variant of SCoNEt with L_1-norm to force the matrices to be sparse, as suggested in [12]. **NAC***: a *cross-domain* model , which is a variant of the proposed NAC model without employing the attention mechanism, thus not weighting the users' preferences across domains. This variant serves as a baseline to evaluate the importance of the proposed cross-domain behavioral attention mechanism in the proposed NAC model.

4.4 Performance Evaluation

Tables 3 and 4 present the experimental results in terms of recall and NDCG, respectively. The cross-domain models CLFM, SCoNet, NAC* and NAC significantly outperform the single-domains models BPR and NeuMF. This is obtained by exploiting users' preferences in the source domains when generating recommendations, thus reducing the data sparsity in the target domain. The proposed NAC model achieves an 6.63% improvement on average in terms of recall when comparing with the second best method of SCoNet. Similarly, NAC outperforms SCoNet by an average improvement of 8.18% in terms of NDCG, for all the cross-domain recommendation tasks. Using the paired t-test we found that NAC is superior over all the competitive approaches for p <0.05.

NAC beats the baselines, as it adaptively selects the weights of users' preferences when transferring the knowledge of users' behaviors from the source domains to the target one. Consequently, the proposed NAC model can self-adjust the subset users that express similar behavior across domains, thus producing accurate cross-domain recommendations. On the other hand, the cross-domain CLFM model uses a subspace learning strategy to linearly associate users' preferences in a common latent space, thus not capturing users' complex preferences across the domains. Although the most competitive method of SCoNet can capture users' different behaviors based on the joint learning approach of stich networks, a weighting strategy is omitted which explains its limited performance, compared to the proposed NAC model.

Table 4: Effect on NDCG. Bold values denote the best scores, using the paired t-test (p <0.05). The last column expresses the relative improvement of the proposed NAC model, compared to the second best method.

Target domain	BPR	NeuMF	CLFM	SCoNet	NAC*	NAC	Improvement (%)
Baby Care	0.2035	0.2221	0.2846	0.3296	0.3086	**0.3675**	**11.49**
Books	0.0940	0.1045	0.1214	0.1694	0.1539	**0.1792**	**5.75**
Destinations	0.2899	0.3368	0.3644	0.4139	0.3922	**0.4587**	**10.82**
Music	0.1124	0.1305	0.1712	0.1992	0.1904	**0.2139**	**7.37**
Online Stores	0.1672	0.1887	0.2894	0.3292	0.3133	**0.3577**	**8.64**
Personal Care	0.1285	0.1426	0.1923	0.2294	0.2109	**0.2491**	**8.58**
Sport/Outdoor	0.1167	0.1262	0.1425	0.1661	0.1559	**0.1725**	**3.85**
Toys	0.1618	0.1829	0.2597	0.2783	0.2658	**0.3017**	**8.40**
Used Cars	0.0851	0.0962	0.1466	0.1747	0.1702	**0.1930**	**10.42**
Video/DVD	0.2313	0.2662	0.3410	0.4093	0.3825	**0.4359**	**6.49**

To further verify this, we observe that the NAC* variant has limited performance when comparing with the NAC model, having relative drops of -10.23% and -12.89% on average in terms of recall and NDCG, respectively. This indicates that indeed the cross-domain behavioral attention mechanism of the proposed NAC model is a key factor to boost the recommendation accuracy, by adaptively assigning larger weight to the subset of users that have similar behavior across the different domains.

4.5 Cold-Start Analysis

In the next set of experiments, we study the performance of the examined models on the cold-start scenario. We define users with less than five appearances in the training set of the target domain as cold-start users. In Tables 5 and 6 we report the effect on recall and NDCG for cold-start users, respectively.

In addition, for each measure we report the average relative drop of each examined model for cold-start users, compared to the performance on all users as presented in Tables 3 and 4 in terms of recall and NDCG, respectively. We observe that for all models there is a drop on both measures in the cold-start case. In particular, single-domain models that are only trained on users' preferences in the target domain, that is the BPR and NeuMF models, have a significant drop of recall and NDCG in the ranges of $22.71 - 31.94\%$ and $22.64 - 31.87\%$, respectively.

The cross-domain models of CLFM, SCoNet, NAC* and NAC are less influenced by cold-start users. In particular, as shown in Tables 5 and 6, all cross-domain models can downsize the negative effect of cold-start users on the performance, with the drops of recall and NDCG being in the ranges of $11.95 - 24.35\%$ and $15.02 - 20.46\%$. This occurs because the cross-domain models exploit the selections of users in multiple domains, hence the cold-start problem has relatively less impact on the models' performances. Evaluated against the second best method of SCoNet, we note that the proposed NAC model maintains relatively high the quality of recommendations, achieving an average improvement of 10.24% and 8.52% in terms of recall and NDCG, for $^*p$ <0.05. This is very crucial in recommendation systems, as in real-world applications there are often many inactive or new users with poor history record, corresponding to cold-start users.

5 CONCLUSIONS

We presented a neural attentive model for generating cross-domain recommendations. The key idea of our NAC model is to transfer the knowledge of users' preferences across domains, considering the cross-domain latent effects of multiple source domains on users' selections in a target domain. Consequently, NAC captures users' complex preferences in different domains. In addition, we introduce a cross-domain behavioral attention mechanism to adaptively perform the weighting of users' preferences from the source domains, and therefore focus on a subset of users that have similar behaviors across domains. Our experiments showed that the proposed approach significantly outperforms baseline methods, proving the importance of our neural architecture and cross-domain behavioral attention mechanism for both ordinary and cold-start users. In fact, we observed that the cross-domain behavioral attention mechanism plays a crucial role in boosting the cross-domain recommendation accuracy by at least 10%. This means that the weights' self-adaptation of users' different preferences across various domains is a key factor in cross-domain strategies. As future work, we plan to extend the proposed NAC model for Location-based Social Networks [1], as well as explore ways to generate social-based recommendations [22], boost information spread [2], perform social event detection [24] and model preference dynamics [23].

REFERENCES

[1] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. 2017. Personalized Keyword Boosting for Venue Suggestion Based on Multiple LBSNs. In *Proceedings of ECIR*. 291–303.

[2] Stefanos Antaris, Dimitrios Rafailidis, and Alexandros Nanopoulos. 2014. Link injection for boosting information spread in social networks. *Social Netw. Analys. Mining* 4, 1 (2014), 236.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR*.

[4] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Distributed collaborative filtering with domain specialization. In *Proceedings of ACM RecSys*. 33–40.

[5] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of RecSys*. 39–46.

[6] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. 2011. Cross-Domain Recommender Systems. In *Proceedings of ICDMW*. 496–503.

[7] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. In *Proceedings of SIGIR*. 515–524.

[8] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. 2013. Cross-Domain Recommendation via Cluster-Level Latent Factor Model. In *Proceedings of ECML PKDD*. 161–176.

[9] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. 2013. Laplacian Sparse Coding, Hypergraph Laplacian Sparse Coding, and Applications. *IEEE*

Table 5: Effect on recall for cold-start users. The last row expesses the relative drop of each model, compared to the perfomance on all users in Table 3.

Target domain	BPR	NeuMF	CLFM	SCoNet	NAC*	NAC	Improvement (%)
Baby Care	0.3178	0.3381	0.4293	0.4160	0.4269	**0.4789**	**15.12**
Books	0.0781	0.0899	0.1210	0.1429	0.1377	**0.1582**	**10.73**
Destinations	0.2614	0.2947	0.3829	0.3778	0.3701	**0.4192**	**10.93**
Music	0.1248	0.1314	0.1828	0.1957	0.1926	**0.2233**	**14.06**
Online Stores	0.2212	0.2726	0.3915	0.3954	0.3719	**0.4118**	**4.15**
Personal Care	0.1037	0.1197	0.1597	0.1866	0.1762	**0.2141**	**14.72**
Sport/Outdoor	0.0957	0.0979	0.1297	0.1634	0.1582	**0.1756**	**7.47**
Toys	0.1970	0.2403	0.2827	0.3038	0.3036	**0.3292**	**8.37**
Used Cars	0.0738	0.0851	0.1056	0.1307	0.1284	**0.1409**	**7.80**
Video/DVD	0.2869	0.2966	0.4027	0.4121	0.3850	**0.4494**	**9.06**
Avg. drop (%)	-29.32	-29.15	-16.76	-17.02	-15.61	-14.24	-

Table 6: Effect on NDCG for cold-start users. The last row expesses the relative drop of each model, compared to the perfomance on all users in Table 4.

Target domain	BPR	NeuMF	CLFM	SCoNet	NAC*	NAC	Improvement (%)
Baby Care	0.1386	0.1718	0.2338	0.2778	0.2479	**0.3065**	**10.35**
Books	0.0650	0.0769	0.1032	0.1402	0.1250	**0.1497**	**6.76**
Destinations	0.2075	0.2410	0.2950	0.3417	0.3119	**0.3886**	**13.72**
Music	0.0787	0.0925	0.1475	0.1675	0.1527	**0.1753**	**4.64**
Online Stores	0.1210	0.1439	0.2402	0.2712	0.2625	**0.3037**	**11.97**
Personal Care	0.0950	0.1039	0.1656	0.1953	0.1807	**0.2113**	**7.92**
Sport/Outdoor	0.0798	0.0875	0.1213	0.1381	0.1293	**0.1461**	**5.80**
Toys	0.1208	0.1364	0.2139	0.2359	0.2256	**0.2536**	**7.51**
Used Cars	0.0597	0.0692	0.1225	0.1484	0.1365	**0.1598**	**7.65**
Video/DVD	0.1749	0.1816	0.2791	0.3347	0.3221	**0.3643**	**8.84**
Avg. drop (%)	-28.56	-27.33	-16.36	-16.40	-17.67	-16.15	-

Trans. Pattern Anal. Mach. Intell. 35, 1 (2013), 92–104.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of WWW*. 173–182.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of WWW*. 173–182.

[12] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of CIKM*. 667–676.

[13] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of WWW*. 595–606.

[14] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[15] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction. In *Proceedings of IJCAI*. 2052–2057.

[16] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of KDD*. 1831–1839.

[17] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-Domain Collaborative Filtering with Factorization Machines. In *Proceedings of ECIR*. 656–661.

[18] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2018. A Contextual Attention Recurrent Architecture for Context-Aware Venue Recommendation. In *Proceedings of SIGIR*. 555–564.

[19] Simon Meyffret, Emmanuel Guillot, Lionel Medini, Frederique Laforest Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. 2012. RED: a Rich Epinions Dataset for Recommender Systems. *LIRIS* hal-01010246 (2012).

[20] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-Stitch Networks for Multi-task Learning. In *Proceedings of IEEE, CVPR*. 3994–4003.

[21] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *Proceedings of AAAI*.

[22] Dimitrios Rafailidis and Fabio Crestani. 2016. Collaborative Ranking with Social Relationships for Top-N Recommendations. In *Proceedings of SIGIR*. 785–788.

[23] Dimitrios Rafailidis and Alexandros Nanopoulos. 2015. Repeat Consumption Recommendation Based on Users Preference Dynamics and Side Information. In *Proceedings of WWW - Companion Volume*. 99–100.

[24] Dimitrios Rafailidis, Theodoros Semertzidis, Michalis Lazaridis, Michael G. Strintzis, and Petros Daras. 2013. A Data-Driven Approach for Social Event Detection. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*.

[25] Steffen Rendle. 2012. Factorization Machines with libFM. *ACM TIST* 3, 3 (2012), 57:1–57:22.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of UAI*. 452–461.

[27] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive Recurrent Social Recommendation. In *Proceedings of SIGIR*. 185–194.

[28] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *Proceedings of KDD*. 2309–2318.

[29] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *Proceedings of IJCAI*. 3119–3125.

[30] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of ICML*. 2048–2057.

[31] Lie Xu, Chiu-sing Choy, and Yi-Wen Li. 2016. Deep sparse rectifier neural networks for speech denoising. In *Proceedings of the IEEE International Workshop on Acoustic Signal Enhancement*. 1–5.