Cross-domain Recommendation with Bridge-Item Embeddings

CHEN GAO and YONG LI, Tsinghua University
FULI FENG, National University of Singapore
XIANGNING CHEN and KAI ZHAO, Tsinghua University
XIANGNAN HE, University of Science and Technology of China
DEPENG JIN, Tsinghua University

Web systems that provide the same functionality usually share a certain amount of items. This makes it possible to combine data from different websites to improve recommendation quality, known as the *cross-domain recommendation* task. Despite many research efforts on this task, the main drawback is that they largely assume the data of different systems can be *fully shared*. Such an assumption is unrealistic different systems are typically operated by different companies, and it may violate business privacy policy to directly share user behavior data since it is highly sensitive.

In this work, we consider a more practical scenario to perform cross-domain recommendation. To avoid the leak of user privacy during the data sharing process, we consider sharing only the information of the item side, rather than user behavior data. Specifically, we transfer the item embeddings across domains, making it easier for two companies to reach a consensus (e.g., legal policy) on data sharing since the data to be shared is user-irrelevant and has no explicit semantics. To distill useful signals from transferred item embeddings, we rely on the strong representation power of neural networks and develop a new method named as NATR (short for ***N**eural **A**ttentive **T**ransfer **R**ecommendation*). We perform extensive experiments on two real-world datasets, demonstrating that NATR achieves similar or even better performance than traditional cross-domain recommendation methods that directly share user-relevant data. Further insights are provided on the efficacy of NATR in using the transferred item embeddings to alleviate the data sparsity issue.

CCS Concepts: • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Cross-domain recommendation, transfer learning, privacy protection

ACM Reference format:
Chen Gao, Yong Li, Fuli Feng, Xiangning Chen, Kai Zhao, Xiangnan He, and Depeng Jin. 2021. Cross-domain Recommendation with Bridge-Item Embeddings. *ACM Trans. Knowl. Discov. Data* 16, 1, Article 2 (June 2021), 23 pages.
https://doi.org/10.1145/3447683

Authors' addresses: C. Gao, Y. Li, X. Chen, K. Zhao, and D. Jin, Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; emails: gc16@mails.tsinghua.edu.cn, liyong07@tsinghua.edu.cn, {cxn15, k-zhao16}@mails.tsinghua.edu.cn, jindp@tsinghua.edu.cn; F. Feng, National University of Singapore, Computing 1, Computing Drive, Singapore 117417; email: fulifeng93@gmail.com; X. He, University of Science and Technology of China, 443 Huangshan Road, Hefei, China 230027; email: xiangnanhe@gmail.com.

1556-4681/2021/06-ART2 $15.00
https://doi.org/10.1145/3447683

1 INTRODUCTION

In the current Web ecosystem, it is common that some websites have a certain degree of homogeneity in their functionality and provided information. For example, there are many overlapped hotels on Trip.com and Booking.com, overlapped movies on IMDb and Douban,[1] and overlapped products on Amazon and eBay. From the perspective of building recommendation services, it means that the models for such two homogeneous domains are dealing with many items that are the same. This provides opportunities to improve the recommendation quality by enriching data. For example, if domain A does not have sufficient data on some items (*i.e.*, sparse or cold-start items) while the other domain B does have, e.g., a movie first released in the US may have many ratings on IMDb but not on Douban, then the recommendation for these items on domain A can be potentially improved by integrating the data of domain B. The task of leveraging auxiliary data from other domains to improve recommendation quality of the target domain is known as the *cross-domain recommendation* [1–6].

Existing works on cross-domain recommendation have primarily focused on directly aggregating data from multiple domains [1–5]. In other words, these methods assume that during model training of the target domain, user behavior data of other domains are directly accessible. For example, the representative **Collective Matrix Factorization (CMF)** [3] method extends **Matrix Factorization (MF)** by jointly learning user embeddings and item embeddings from the user–item interaction matrix of multiple domains. Despite effectiveness, the assumption that user behavior data can be fully shared across domains is questionable. Typically, different domains (websites) are operated by different companies, and thus it is difficult to let them share user behavior data due to the constraint of company policy.

In this work, we aim to provide a more realistic solution for cross-domain recommendation. To avoid any chance of leaking user privacy, we abandon the sharing of user-relevant data, neither behavior logs nor demographic attributes. However, this will pose challenges to transfer the **collaborative filtering (CF)** signal from one domain to another one, since CF is typically modeled through mining user–item interaction data (e.g., user purchase and click logs). To address this technical challenge, we propose to share the item embeddings, which are learned by reconstructing the user–item interaction matrix. The advantages are two-fold: (1) item embeddings can still encode certain CF signal by reflecting item similarities based on user behaviors (e.g., which items are frequently co-rated by users)[2]; and (2) item embeddings are latent vectors that have no explicit semantics; as such, the risk of leaking user privacy can be kept to a minimum, which makes it easier for two companies to reach a legal policy for data sharing.[3]

Our proposed solution, which has three steps, is illustrated in Figure 1. In the first step, an embedding-based recommender model , MF for example, is trained on the user–item interaction matrix of the auxiliary domain to obtain item embeddings. In the second step, item embeddings of the auxiliary domain are sent to the target domain; note that only the embeddings of overlapped items are necessary to be sent, which are subjected to the data-sharing policy between two companies. Finally, the target domain trains a recommender model with the consideration of the transferred item embeddings. The first two steps are straightforward to implement, and the main challenges lie in how to design a model to effectively incorporate transferred item embeddings in the last step. We summarize the key challenges as follows.

[1] Douban.com is a Chinese website that allows registered users to rate movies, music, books, and the like.
[2] A representative example of using the signal in item embeddings for recommendation is the item-based CF methods [7–9].
[3] https://www.nytimes.com/interactive/2018/06/03/technology/facebook-device-partners-users-friends-data.html.

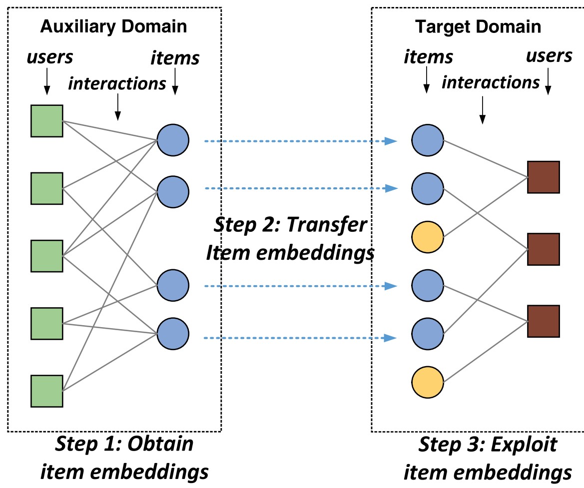


Fig. 1. Illustration of our solution for cross-domain recommendation without sharing user-relevant data.

— **Unclear predictive signal of transferred item embeddings.** It is unclear whether and which transferred item embeddings contain useful signals in estimating a user's preference on an item in the target domain. Note that one motivation for conducting cross-domain recommendation is to alleviate the data sparsity problem in the target domain. However, the data sparsity problem may also exist in the auxiliary domain for some items, or the other way round the data in the target domain is already sufficient and does not require the extra supplement. As such, it is challenging to distill useful signal from the transferred item embeddings and integrate them into the predictive model of the target domain.
— **Varying importance of transferred item embeddings.** As mentioned, the data of the auxiliary domain is not oracle—it is likely that user behaviors on some items are sparse and are insufficient to learn good embeddings for them. As such, it is a common case that the quality of item embeddings varies, where items with many interactions may have good quality and vice versa. Since it is already difficult to judge the quality of learned item embeddings for the auxiliary domain, it becomes even more challenging for the target domain to utilize such unknown- and varied- quality item embeddings well.
— **Embedding dimension discrepancy in latent space.** The data for training in the two domains may be of different scales and have different distributions. Therefore, the optimal embedding size for the two domains may be different. As such, existing cross-domain recommendation solutions that perform regularization on embedding matrices will fail [10]. Moreover, even though we restrict their embedding sizes to be the same, the semantics of their embedding dimensions are different and cannot be directly aligned.

To solve the above-mentioned challenges, we design a novel model named **Neural Attentive-Transfer Recommendation (NATR)**. Briefly, our proposed method relies on the strong representation power of neural networks and discriminative power of attention mechanisms to leverage the transferred item embeddings. Specifically, we design (1) a domain-level attention unit to dynamically adjust the importance of the predictive signal of the two domains; (2) an item-level attention unit to determine which embeddings of transferred items are more useful in constructing user representation for further prediction; and (3) a domain adaption layer to bridge the discrepancy between the embedding space of the two domains. By tailoring our solution for addressing the highlighted challenges, our NATR method demonstrates its strong performance in cross-domain recommendation, and meanwhile preserves user privacy during data sharing.

To summarize, the main contributions of this work are as follows.

— We present a new paradigm for cross-domain recommendation without sharing user-relevant data, in which only item-side data can be shared across domains. To allow the transferring of CF signal, we propose to share the item embeddings which are learned from user-item interactions of the auxiliary domain.
— We propose a new solution NATR to resolve the key challenges in leveraging transferred item embeddings. The two-level attention design allows NATR to distill useful signal from transferred item embeddings, and appropriately combine them with the data of the target domain.
— We conduct extensive experiments on two real-world datasets to demonstrate our proposed method. More ablation studies verify the efficacy of our designed components and the utility of transferred item embeddings in addressing the data sparsity issue.

The remainder of this article is as follows. Compared with the conference version [11], this article goes much deeper both in technical contribution and experimental evaluation. We first formulate the research problem in Section 2. We then elaborate our proposed method in Section 3. We conduct experiments in Section 4, before discussing related work in Section 5. Lastly, we conclude this article in Section 6.

2 PROBLEM FORMULATION

We first introduce some notations used in the article. We represent matrices, vectors, and scalars as bold capital letters (*e.g.,* $\mathbf{X}$), bold lower letters (*e.g.,* $\mathbf{x}$), and normal lowercase letters (*e.g.,* x), respectively. If not otherwise specified, all vectors are in a column form; $\mathbf{X}^T$ denotes the transpose of $\mathbf{X}$. We use symbols σ, $ReLU$, and $\odot$ to denote the *sigmoid* function, rectifier function, and element-wise production operation, respectively.

2.1 Cross-domain Recommendation

A typical problem setting of *cross-domain recommendation* is leveraging the data from an *auxiliary* domain to facilitate the recommendation quality in a *target* domain with overlapped items. In the *target domain*, where M and N denote the number of users and items, respectively, we have a user-item interaction matrix $\mathbf{Y}^t \in \mathbb{R}^{M\times N}$ with a binary value at each entry defined as,

$$y_{ui}^t = \begin{cases} 1, & \text{if } u \text{ has interacted with } i; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Similarly, in the *auxiliary domain*, we have another binary user–item interaction matrix $\mathbf{Y}^a \in \mathbb{R}^{K\times L}$, where K and L are the number of users and items. Note that a portion of L items also occurs in the target domain, which are named as *bridge items*. From the interaction matrices $\mathbf{Y}^t$ and $\mathbf{Y}^a$, the goal of cross-domain recommendation is to learn a predictive function to estimate the likelihood that a given user u will interact with item i in the target domain.

2.2 Cross-domain Recommendation Without Sharing User-relevant Data

Distinct from the typical problem settings of cross-domain recommendation, we abandon the direct sharing of user behavior data (the user–item interaction matrix $\mathbf{Y}^a$ in the auxiliary domain). This is because directly sharing user behavior data may violate the business privacy policy of different companies operating the auxiliary and target domains. Instead, we propose a solution that only transfers the embeddings of bridge items which are offline learned in the auxiliary domain, as illustrated in Figure 1. We define the transferred item embedding matrix $\mathbf{Q}^a = [\mathbf{q}_1^a, \ldots, \mathbf{q}_N^a] \in \mathbb{R}^{D' \times N}$ as,

$$\mathbf{q}_i^a = \begin{cases} \tilde{\mathbf{q}}_i^a, & \text{if item } i \text{ is a bridge item;} \\ \mathbf{0}, & \text{otherwise;} \end{cases} \tag{2}$$

where $\mathbf{0} \in \mathbb{R}^{D'}$ is an all-zero vector and $\tilde{\mathbf{q}}_i^a \in \mathbb{R}^{D'}$ is the offline learned embedding of item i in the auxiliary domain. It should be noted that we organize the transferred item embeddings in $\mathbf{Q}^a$ in the same order of item IDs in the target domain to enable looking up an item embedding with its ID. Here we assume the availability of $\tilde{\mathbf{q}}_i^a$, *i.e.,* the company operating the auxiliary domain has employed an embedding-based recommendation system [12]. Note that the assumption is practical since embedding-based recommendation solutions are widely applied in the industry [13, 14].

After introducing the transferred item embeddings $\mathbf{Q}^a$ from the auxiliary domain, we formulate the problem of cross-domain recommendation without sharing user-relevant data as follows.

Input: The user–item interaction data in the target domain $\mathbf{Y}^t$, and the transferred item embeddings $\mathbf{Q}^a$ from the auxiliary domain.

Output: A predictive model to estimate the likelihood that a user u will interact with an item i in the target domain. Specifically, taking u, i, and $\mathbf{y}_u^t$ which is the interaction history[4] of u in the target domain, as input, the model has to predict,

$$\hat{y}_{ui}^t = f(u, i, \mathbf{y}_u^t), \tag{3}$$

where $\hat{y}_{ui}^t \in [0, 1]$ denotes the probability of interaction between user u and item i.

After obtaining the predictive model, we can use it to score all items for a given user u, and select the top-ranked (*i.e.,* with higher interaction probability) items as the recommendation results for u. It should be noted that there indeed exist user and item attributes in both the auxiliary and target domains. However, to simplify the scenario of the cross-domain recommendation task, we only emphasize the user–item interactions, which is a common setting of existing works [2, 10].

3 PROPOSED METHOD

To solve the problem of cross-domain recommendation without sharing user-relevant data, we rely on the strong representation ability of neural networks and devise a new solution, named, NATR, exploiting the transferred item embeddings. Figure 2 illustrates the architecture of our proposed NATR model, which are made up of the following four layers.

- **Transfer-enhanced Embedding Layer.** We project sparse user and item representations into dense vectors. A dimension-adaption module is adopted to solve the dimension discrepancy problem of transferred item embeddings.
- **Item-level Attentive Layer.** To enrich user representations, we fuse the transferred embedding of items a user interacted with to an *additional user embedding* with an item-level attention unit to model the varying importance of items.
- **Domain-level Attentive Layer.** With consideration of the diversity across domains, we make use of a domain-level attention unit to control the influence of predictive signals from two domains.
- **Prediction Layer.** Finally, we utilize an inner-product model as the predictive function since our work mainly focuses on devising a framework to exploit the transferred item embeddings.

In the following, we elaborate on the details of the aforementioned four layers.

[4]Note that $\mathbf{y}_u^t$ is the transpose of the u-th row of $\mathbf{Q}^t$.

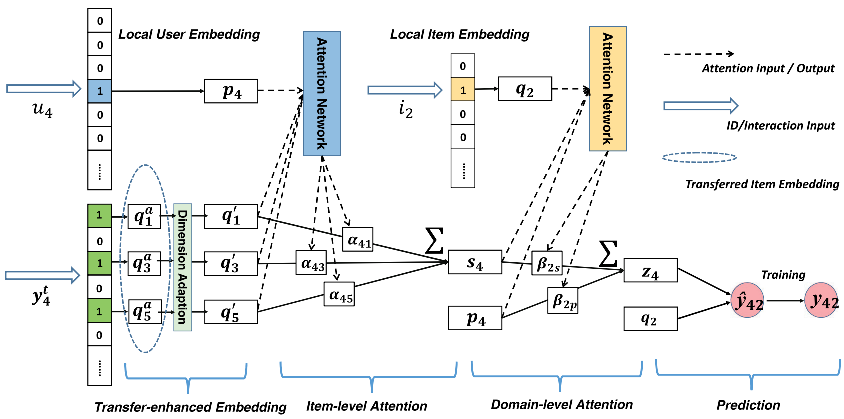


Fig. 2. The architecture of our proposed NATR model. Here we take the prediction of user–item pair (4, 2) as an example; user 4 has interacted with item 1, 3, and 5 in the target domain; then we utilize $\mathbf{q}_1^a$, $\mathbf{q}_3^a$, and $\mathbf{q}_5^a$ transferred from the auxiliary domain to assist predicting $\mathbf{y}_{42}$ in the target domain. Note that $\mathbf{q}^a$ is the only available auxiliary data in the target domain. Best view in color.

3.1 Transfer-enhanced Embedding Layer

Latent factor model (LFM) is one kind of general framework in collaborative recommender systems, which associates each user and item with real-valued vectors. Considering that LFMs have achieved success in a wide range of recommendation tasks [7, 15, 16], we project sparse user and item representations into real-valued vectors. Specifically, we first encode user ID (u) and item ID (i) into one-hot encodings as follows,

$$\mathbf{v}_u^U = \text{one-hot}(u), \mathbf{v}_i^I = \text{one-hot}(i), \tag{4}$$

where $\mathbf{v}_u^U$ ($\mathbf{v}_i^I$) $\in \mathbb{R}^N$ is a vector with all zero values except the u-th (i-th) entry with value 1. We then project the sparse one-hot encodings ($\mathbf{v}_u^U$ and $\mathbf{v}_i^I$) and multi-hot interaction history ($\mathbf{y}_u^t$) to *local embeddings* and *transferred embeddings*, respectively.

Local embeddings. To project the one-hot user (item) encoding, we employ an embedding layer, which is defined as a fully connected layer without bias term as follows,

$$\mathbf{p}_u = \mathbf{P}^T \mathbf{v}_u^U, \quad \mathbf{q}_i = \mathbf{Q}^T \mathbf{v}_i^I, \tag{5}$$

where $\mathbf{P} \in \mathbb{R}^{N \times D}$ and $\mathbf{Q} \in \mathbb{R}^{M \times D}$ are the parameters to be learned. The obtained embeddings $\mathbf{p}_u$ and $\mathbf{q}_i \in \mathbb{R}^D$ are named as *local embeddings* since they are learned merely with information from the target domain.

Transferred embeddings. In our problem, transferred embeddings of bridge items are the only auxiliary data accessible in the target domain. From the perspective of representative learning, there are two manners to leverage these item embeddings in collaborative filtering: user-based [15, 16] and item-based [7, 8]. Specifically, when predicting the probability that user u will choose item i in the target domain, user-based CF means directly combining embeddings of i of two domains to match $\mathbf{p}_u$ while item-based manner means matching $\mathbf{q}_i$ with transferred embedding of user's historically interacted items. There are two key aspects to make item-based manner a more convincing choice. First, when item i is not a bridge item, user-based CF solution cannot bring any help to prediction. Second, user-based CF can only distill implicit preferences while neglecting the explicit preferences of a user (*i.e.*, the historically interacted items), while item-based CF can extract explicit preferences through leveraging transferred embeddings of bridge items and multi-hot encoding of historical interactions to enrich user representation.

Therefore, we look up the transferred embedding $\mathbf{q}_j^a$ from $\mathbf{Q}^a$ for each item j with $y_{uj}^t = 1$. Note that $\mathbf{q}_j^a$ will be an all-zero vector if item j is not a bridge item (see Equation (2) for details). As mentioned above, there may exist the challenge of *embedding dimension discrepancy* across two domains, *i.e.*, $D \neq D^{'}$, since the training data in two domains may be of different scale and have different distributions. To solve this problem, we employ a fully-connected layer to adapt the dimension of transferred embeddings, which is formulated as follows,

$$\mathbf{q}_j^{'} = \mathbf{W}_0^T \mathbf{q}_j^a + \mathbf{b}_0, \tag{6}$$

where $\mathbf{W}_0 \in \mathbb{R}^{D^{'} \times D}$ and $\mathbf{b}_0 \in \mathbb{R}^D$ are learnable parameters of mapping matrix and bias. Note that $\mathbf{W}_0$ and $\mathbf{b}_0$ are the only parameters here as the transferred embedding matrix $\mathbf{Q}^a$ is learned offline in the auxiliary domain.

With the above neural components, we project the sparse one-hot and multi-hot encodings into local user and item embeddings, $\mathbf{p}_u$ and $\mathbf{q}_i$, and transferred item embeddings, $\{\mathbf{q}_j^{'} | y_{uj}^t = 1\}$. We introduce our item-based CF solution of leveraging transferred item embeddings detailedly in the following parts.

3.2 Item-level Attentive Layer

The key objective of embedding-based recommendation model is to capture relation between user and item in the latent space [17], therefore it is critical to explicitly build the relation between transferred item embeddings and local user embedding in our problem. As mentioned above, motivated by item-based CF [7, 8], which encodes the historical interaction behaviors of a user to enrich the user representation, we fuse the transferred item embeddings $\{\mathbf{q}_j^{'} | y_{uj}^t = 1\}$ into an *additional user embedding* $\mathbf{s}_u$. Besides historical interactions, $\mathbf{s}_u$ also contains CF signals transferred from the target domain, which can further enhance user representation. Our first inspiration to calculate $\mathbf{s}_u$ is average pooling, a widely used modeling component in neural networks, formulated as follows,

$$\mathbf{s}_u = \frac{\sum_{\{j | y_{uj}^t = 1\}} \mathbf{q}_j^{'}}{|\mathbf{y}_u^t|}, \tag{7}$$

where $|\mathbf{y}_u^t|$ is the l_1-norm of vector $\mathbf{y}_u^t$, which equals to the number of items user u has interacted with.

However, as mentioned above, there exists another key challenge of *varying importance of embeddings*. Considering that different interacted items have embeddings with varying quality and varying importance to represent the preference of a given user, such naive operation may not work well in the real scenario. Therefore, we apply a non-uniform coefficient when fusing the transferred item embeddings:

$$\mathbf{s}_u = \frac{\sum_{\{j | y_{uj}^t = 1\}} \alpha_{uj} \mathbf{q}_j^{'}}{|\mathbf{y}_u^t|}. \tag{8}$$

To model the various item importance in a user-sensitive fashion, here we introduce attention mechanism, which has achieved great success in recommendation tasks [8, 18]. Specifically, the item-level attention unit learns a specific weight α_{uj} for every transferred item embedding $\mathbf{q}_j^{'}$

according to the following formulation,

$$\alpha_{uj} = \frac{e^{a_{uj}}}{\sum_{\{k|y_{uk}^t=1\}} e^{a_{uk}}},$$
$$a_{uj} = \mathbf{w}_1^T ReLU(\mathbf{p}_u \odot \mathbf{q}_j^{'}) + b_1, \tag{9}$$

where $\mathbf{w}_1 \in \mathbf{R}^D$ and b_1 denote the weight matrix and bias of a fully connected layer. The input of the item-level attention unit is the interaction between the user and target item, which makes the learned attention score sensitive to the given user.

3.3 Domain-level Attentive Layer

After obtaining the *local* and *additional* user embeddings, our task become learning a prediction function based on these three embeddings. Different from traditional recommendation models, here we have an extra user embedding. To exploit two user embeddings, we fuse them into a *unified embedding*. This is inspired by some cross-domain recommendation models [2, 19], which have demonstrated that fusing embedding vector learned from multi-modal data is a simple yet effective way to combine signals. Another option is to separately estimate the interaction probability with the two embeddings and fuse the predictions (late fusion). Here we employ early fusion that merges embeddings, allowing us to capture the interaction between two embeddings explicitly. The *unified embedding* via fusion can be denoted as,

$$\mathbf{z}_u = \beta_{si}\mathbf{s}_u + \beta_{pi}\mathbf{p}_u,\ s.t.,\ \beta_s + \beta_p = 1, \tag{10}$$

where β_{si} and β_{pi} are learnable weights for $\mathbf{s}_u$ and $\mathbf{p}_u$, respectively. The aim of β_{si} and β_{pi} is to balance the information from auxiliary and target domain regarding the target item i. In other words, these two weights are item-sensitive. We devise such design to address the key challenge of *unclear predictive signal* which has been mentioned before. That is, in real scenario evaluating different items needs varying amount of auxiliary information.

$$\beta_{si} = \frac{e^{b_{si}}}{e^{b_{si}} + e^{b_{pi}}},\ \beta_{pi} = \frac{e^{b_{pi}}}{e^{b_{si}} + e^{b_{pi}}},$$
$$b_{si} = \mathbf{w}_2 ReLU(\mathbf{s}_u \odot \mathbf{q}_i) + b_2, \tag{11}$$
$$b_{pi} = \mathbf{w}_2 ReLU(\mathbf{p}_u \odot \mathbf{q}_i) + b_2,$$

where $\mathbf{w}_2 \in \mathbb{R}^D$ and b_2 are the parameters of the attention network. Note that the input of the attention network is the interaction between user ($\mathbf{s}_u/\mathbf{p}_u$) and item embeddings, which enables the learned attention scores to be sensitive to item i.

3.4 Prediction Layer

After the operation in aforementioned layer, our problem further turns to predict user interaction with two embeddings: unified user embedding vector $\mathbf{z}_u$ and item embedding vector $\mathbf{q}_i$. Here we adopt a predictive function to estimate y_{ui}^t which is the interaction probability between a given pair of user and item (u, i). Since our work mainly focuses on a general framework for cross-domain recommendation without sharing user-relevant data, we adopt a simple but widely-used inner product model, to estimate the value of y_{ui}^t, which is formulated as

$$\hat{y}_{ui}^t = \sigma({\mathbf{z}_u}^T \mathbf{q}_i), \tag{12}$$

where σ is the *sigmoid* function. Note that this predictive function can be easily extended to more complicated ones, such as the multi-layer perceptron in [20].

To conclude, with three specially devised layers and a prediction layer, the aforementioned three key challenges of our problem are addressed one by one.

3.5 Training

Objective Function. Following the probabilistic optimization framework [15, 20], we first define the likelihood function for an implicit interaction as follows,

$$P_r = \prod_{(u,i)\in\mathcal{Y}_+^t} \hat{y}_{ui}^t \prod_{(u,i)\in\mathcal{Y}_-^t} (1-\hat{y}_{ui}^t), \tag{13}$$

where $\mathcal{Y}_t^+$ denotes the set of observed interactions in interaction matrix of the target domain $\mathbf{Y}^t$ (entries with value of 1), and $\mathcal{Y}_-^t$ denotes negative instances sampled from the unobserved interactions in $\mathbf{Y}^t$ (entries with value of 0). We further take the negative logarithm of the joint probability, and obtain the loss function (*a.k.a, logloss* [21]), which is widely used to optimize recommendation systems with implicit feedbacks [7, 16, 20], to be minimized as follows,

$$\mathcal{L} = -\left(\sum_{(u,i)\in\mathcal{Y}_+^t} \log \hat{y}_{ui}^t + \sum_{(u,i)\in\mathcal{Y}_-^t} \log(1-\hat{y}_{ui}^t)\right). \tag{14}$$

To prevent over-fitting, we adopt l_2 regularization on the parameters in the proposed neural model and obtain the overall objective function,

$$\Gamma = \mathcal{L} + \lambda \sum_{\Theta\in\{\mathbf{P},\mathbf{Q},\mathbf{W}_0,\mathbf{b}_0,\mathbf{w}_1,b_1,\mathbf{w}_2,b_2\}} \|\Theta\|_F^2. \tag{15}$$

Mini-batch Training. We adopt stochastic gradient descent, a widely generic solver for neural models, to optimize our proposed NATR model in the mini-batch mode. To construct a mini-batch, we first sample a batch of historical user–item interaction pairs (u,i). For each (u,i), we then adopt a negative sampling technique [16], which is widely used to handle implicit feedbacks in existing researches [15, 20], to randomly select unobserved items $\{i_1^{'}, \ldots i_n^{'}\}$ for user u with a sampling ratio of n. Note that some works [22] also proposed to optimize the recommendation model without negative sampling. However, for our proposed model, negative sampling is a more suitable choice for optimization. After the sampling, we obtain n triplets $\{(u,i,i_1^{'}), \ldots, (u,i,i_n^{'})\}$ for each instance in the batch. With the constructed mini-batch, we take a gradient step to minimize the objective function.

3.6 Discussion

3.6.1 Potential Attackers. Here we discuss the potential privacy leakage if user embeddings are shared. Our proposed framework, which avoids sharing user embeddings, can defend the attackers that want to infer private information from the user embeddings. In fact, the attackers can reveal two kinds of user privacy, interacted items, and user profiles. Specifically, for the first kind, a user may do not want the target domain's employees infer what he/she has purchased at the auxiliary domain; for the second kind, a user may do not want the target domain's employees to infer his profiles such as age and gender. For the first kind of privacy leakage, an adversarial model can be a recommendation model that tries to match user embeddings with item embeddings. For the second kind of privacy leakage, an adversarial model can be a classification/clustering model that assigns users to certain user profile categories.

It is worth mentioning that existing research based on differential privacy [23], k-anonymity [24], or t-closeness [25] always provide theoretical privacy bounds. In this work, it is hard to analyze the privacy bound of sharing embedding vectors directly, and thus we consider testing the impact on the defense to potential attackers as a feasible solution. We leave it as an important future work.

3.6.2 Platform Privacy. In fact, this work's target is to protect user privacy. The shared item embeddings are low-dimensional representations, which implicitly encode items' features. Some platform-aware information can be inferred or partly inferred based on them. For example, the overlap of products can be calculated. Another example is the embeddings may be used to predict product-related information. Luckily, compared with user privacy, such information is not so critical and sensitive. Here we emphasize that: first, this work's target is user privacy, which is an important concern in existing cross-domain recommendation models; second, platform information is partly leaked indeed, but it is not a big issue.

4 EXPERIMENTS

In this section, we conduct extensive experiments on two real-world datasets to answer the following research questions:

— **RQ1:** How does our proposed NATR model perform compared with the state-of-the-art methods for cross-domain recommendation tasks?
— **RQ2:** Can the proposed NATR alleviate the data sparsity problem in the target domain?
— **RQ3:** What are the effects if we remove item-level and domain-level attention models in our proposed NATR?
— **RQ4:** How does the quality of data in the auxiliary domain affect the recommendation performance of our NATR model?
— **RQ5:** How about the recommendation performance when there are many non-overlapped items?
— **RQ6:** How about the recommendation performance when we choose another kind of design of attention module in NATR?

In what follows, we first describe the experimental settings and then answer the above six research questions.

4.1 Experimental Settings

4.1.1 Datasets. We experiment with two real-world datasets that both contain implicit interactions from two domains.

— **ML-NF Dataset. MovieLens (ML)** and **Netflix (NF)** are two popular platforms with movie recommendation services, in which there are a large portion of overlapped movies. Here we take ML and NF as the *auxiliary* and *target* domains (*i.e.,* our target is to improve the recommendation performance in NF), respectively. We obtain user–movie interactions in ML and NF from two widely used public movie rating datasets.[5,6] Note that we identify movies with the same name in the two datasets as *bridge items*. Here we conduct whole-string matches to avoid wrong matches as possible. By filtering bridge items and their associated ratings,[7] we reserve 5,568 movies, 14,630 ML users, and 31,038 NF users in this dataset. Finally, we

[5] https://grouplens.org/datasets/movielens/.
[6] https://www.kaggle.com/laowingkin/netflix-movie-recommendation/data.
[7] To better evaluate the performance, we follow the setting of previous works [3, 26] and only reserve the bridge items (*i.e.*, neglect items occurring only in one domain). However, our model is also suitable to perform recommendation for those items not overlapped, which has been introduced in detail in Section 3.1, and we will evaluate this in Section 4.6.

Table 1. Statistics of Our Evaluation Datasets.

Dataset	Item#	Auxiliary Domain		Target Domain	
		User#	Rec#	User#	Rec#
ML-NF	5,568	31,038	2,269,179	14,630	152,206
TC-IQI	4,851	35,398	314,621	19,999	78,429

intentionally transform the rating data into binary (1/0 indicate whether a user has interacted with an item or not) to fit the problem setting of implicit feedback [27].

— **TC-IQI Dataset.** This dataset is collected by [2] to evaluate *cross-domain recommendation* performance of online video contents. In this dataset, there are historical interactions between users and videos from two mainstream video websites, **iQiyi (IQI)**[8] and **Tencent Video (TC)**[9] in China. To investigate the performance of facilitating recommendation performance in target domain via leveraging information from auxiliary domain, we regard *IQI* (*TC*) as the *target* (*auxiliary*) domain since interactions in IQI are sparser. Similarly, we filter *bridge items* via exact name matching across videos from these two domains, and only reserve bridge items and interactions associated with them. Note that videos in this dataset are professional production content widely available on multiple websites.

After the above preprocessing steps, we obtain two final datasets for performance evaluation, the statistics of which are summarized in Table 1.

4.1.2 Evaluation Protocols. Following [20], we employ the widely used *leave-one-out* evaluation protocol in the evaluation stage. Similar with [20, 28], given a user in the target domain, we randomly sample 99 items that are not interacted by the user, and each method ranks one test item among the 99 sampled items. We then adopt two metrics, *HR* and *NDCG*, which are widely used in the literature of recommendation [20, 29], to report the ranking performance of each method:

— **HR@K:** ***Hit Ratio*** **(HR)** measures whether the test item is contained by the top-K item ranking list (1 for yes and 0 for no).

— **NDCG@K:** ***Normalized Discounted Cumulative Gain*** **(NDCG)** extends HR by assigning higher scores to the hits at higher positions in the ranking list.

It should be noted that we calculate HR@K and NDCG@K for each test user, and report the average ones over the whole user set.

4.2 Performance Comparison (RQ1)

4.2.1 Baselines. We compare the performance of our proposed NATR with five baselines, which can be divided into two groups: *single-domain* and *cross-domain*. Here *single-domain* methods refer to those which are merely trained with data from the target domain, while *cross-domain* methods jointly consider the data from both the target and auxiliary domains.

The compared *single-domain* methods are introduced as follows:

— **PMF [16].** ***Probabilistic Matrix Factorization*** **(PMF)** is a MF-based model which exploits negative sampling to handle implicit interaction data. It adopts *logloss* as the loss function and samples several negative items with a ratio when a positive item is fed for training. We tune the learning rate and regularizer and report the best testing performance.

[8]https://www.iqiyi.com.
[9]https://v.qq.com.

- **GMF.** ***Generalized Matrix Factorization*** **(GMF)** is one of the variants of Neural Collaborative Filtering [20], which is the state-of-the-art solution for recommendation tasks with implicit feedback. This method assigns various weights for different dimensions in the dot-product prediction function, which can be regarded as a generalization of vanilla MF. We optimize this model and tune its associated hyper-parameters similarly with the article.
- **NGCF [30].** ***Neural Graph Collaborative Filtering*** **(NGCF)** is the new state-of-the-art collaborative filtering model that adopts graph neural networks to extract high-order connectivity on the user-item graph. We optimize this model and tune its associated hyper-parameters following the original article.
- **NATR-local.** As mentioned in Section 3.1, our NATR model utilizes an item-based CF to leverage transferred item embeddings. Therefore, it is still questionable whether the item-based CF is the only component to improve performance while transferred item embeddings do not help. To demonstrate the effectiveness of transferred embeddings, we degenerate the NATR via adopting local item embeddings rather than transferred item embeddings in the item-level attention unit. Therefore, it is a kind of *single-domain* method. We name it NATR-local and tune it similarly with NATR to report the best performance.

The compared *cross-domain* baselines are as follows.

- **CMF [3].** CMF decomposes the data matrices of multiple interactions simultaneously while sharing embedding vectors of users or items. Here we factorize two interaction matrices from two domains, sharing embedding vectors of those *bridge items*. We carefully tune the weight of two domains, learning rate and regularizer to report the best performance. It is worth mentioning that a recent study on cross-domain recommendation [2] proposed a method named MPF, which adapted vanilla CMF to a special case where all users and items are all overlapped across domains. Apparently, this special setting does not fit our problem, of which only items can be overlapped, and thus regretfully, MPF cannot be adapted to our task.
- **ItemCST [10].** *Coordinate System Transfer* **(CST)** also assumes that both users and items are overlapped and adds two regularization terms in objective functions. Specifically, the two terms set constraints to the embedding distance in two domains for those overlapped users or items. Thus, CST can be adapted to our problem by only reserving item-side regularization term in our task, and we name it as ItemCST. We tune the learning rate and coefficient of regularization term to report the best performance.

  To conclude, CMF is the state-of-art cross-domain recommendation method while facing a high risk of leaking user privacy since it assumes that all interaction data are fully shared. ItemCST is an adapted method from CST, and as it only needs the transferred item embeddings to compute the regularization term of item, it preserves user-relevant data.

We implement the baseline methods and our NATR model in TensorFlow.[10] It should be noted that we set the embedding size of all compared methods to be 64, which is a typical setting in literature [7, 20]. Our primary experiments also demonstrate that 64 is an embedding size with enough ability to represent the user and item.

4.2.2 Parameter Settings. To determine the optimal hyper-parameters of the method, we construct a validation set via randomly selecting an interacted item for each test user, which has not been selected as the test item. During the training phase, we intentionally set the negative sampling ratio as 4 to construct mini-batches with the size of 256 as described in Section 3.5. To optimize the

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

Table 2. Top-K Recommendation Performance Comparison on the ML-NF and TC-IQI Datasets (K is set to 1, 2, 5, 10)

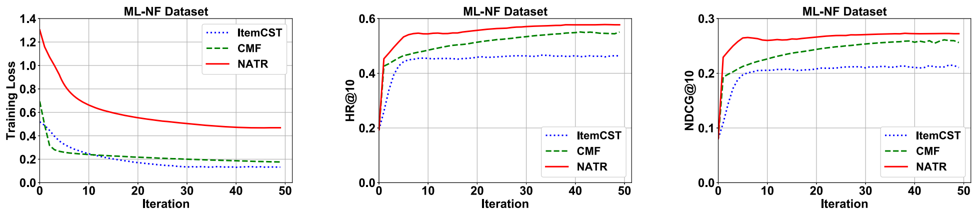
			ML-NF Dataset						
Group	**Method**	**User-relevant Data**	**HR(NDCG)@1**	**HR@2**	**NDCG@2**	**HR@5**	**NDCG@5**	**HR@10**	**NDCG@10**
Cross Domain	**NATR**	**Preserved**	**0.1315**	**0.1976**	**0.1403**	**0.3776**	**0.2110**	**0.5781**	**0.2726**
	ItemCST	Preserved	0.0795	0.1475	0.1005	0.3068	0.1670	0.4846	0.2228
	CMF	Shared	0.1023	0.1903	0.1283	0.3675	0.2025	0.5483	0.2593
Single Domain	NATR-local	Preserved	0.0947	0.1769	0.1253	0.3402	0.1894	0.5183	0.2440
	PMF	Preserved	0.0668	0.1162	0.0796	0.2721	0.1375	0.4494	0.1956
	GMF	Preserved	0.0706	0.1174	0.0816	0.2681	0.1410	0.4284	0.1918
	NGCF	Preserved	0.0961	0.1835	0.1284	0.3470	0.1875	0.5279	0.2461
			TC-IQI Dataset						
Group	**Method**	**User-relevant Data**	**HR(NDCG)@1**	**HR@2**	**NDCG@2**	**HR@5**	**NDCG@5**	**HR@10**	**NDCG@10**
Cross Domain	**NATR**	**Preserved**	**0.2010**	0.2660	**0.2104**	**0.4513**	**0.2881**	0.6035	**0.3365**
	ItemCST	Preserved	0.1161	0.2129	0.1445	0.4194	0.2309	0.6079	0.2904
	CMF	Shared	0.1649	**0.3101**	0.2101	0.4499	0.2668	**0.6595**	0.3326
Single Domain	NATR-local	Preserved	0.1677	0.2552	0.1776	0.4214	0.2412	0.5864	0.2948
	PMF	Preserved	0.0848	0.1238	0.0945	0.2291	0.1326	0.3309	0.1694
	GMF	Preserved	0.1584	0.2445	0.1729	0.4101	0.2425	0.6021	0.3029
	NGCF	Preserved	0.1692	0.2609	0.1772	0.4280	0.2485	0.5932	0.3095

NATR model, we employ the Adagrad optimizer and search its learning rate within {0.001, 0.002, 0.005, 0.01}. In addition, we tune the λ in Equation (15), which balances the loss and regularization terms, in {1e-2, 1e-3, 1e-4,1e-5,1e-6}. As mentioned before, ItemCST and our NATR only rely on the transferred item embeddings as auxiliary data, and in this article, without loss of generality, we adopt PMF in the auxiliary domain and carefully tune its learning rate and regularizer to obtain item embeddings.

We first compare the top-K recommendation performance with baseline methods. We investigate the top-K performance with K setting to $\{1, 2, 5, 10\}$.[11] As described in the evaluation protocols, we test the performance of a ranking list with 100 items. As such, it is reasonable to choose a relatively small K [20]. For every method, we carefully tune the hyper-parameters to report the best performance. To make the results stable and convincing, for each experiment, we run five repetitive instances and report the average values. In Table 2, we report the top-K recommendation performance for the two utilized real-world datasets. We compare our proposed NATR method with three single-domain baselines and two cross-domain ones. From these results, we have the following observations:

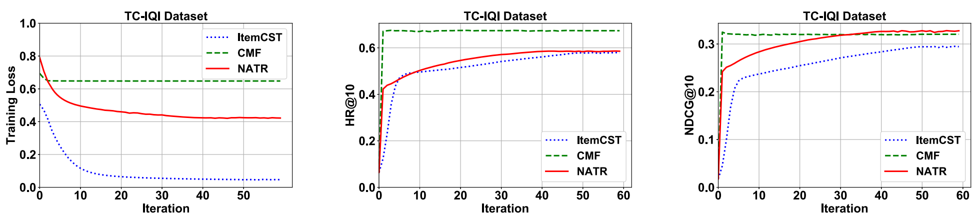
— **NATR significantly improves recommendation performance in the target domain.** (1) For those *single domain* methods that are trained with only interaction data from the target domain, the recommendation performance is relatively poor. PMF achieves the worst performance, which can be explained as the limited representation ability of MF model. (2) NATR-local, a degenerative model of the proposed NATR, outperforms PMF and GMF *w.r.t.* all metrics on the ML-NF dataset and most metrics on the TC-IQI dataset, which justifies the effectiveness of explicitly encoding users' historical interactions. In fact, this can be explained as the ability of the item-based CF component introduced in

[11] Note that HR@K equals to NDCG@K when setting $K = 1$.



(a) Training Loss (b) HR@10 (c) NDCG@10

Fig. 3. Training loss and testing performance of NATR, ItemCST, and CMF in each iteration on ML-NF Dataset.

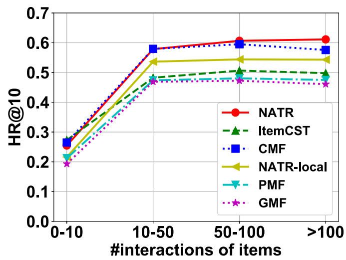


(a) Training Loss (b) HR@10 (c) NDCG@10

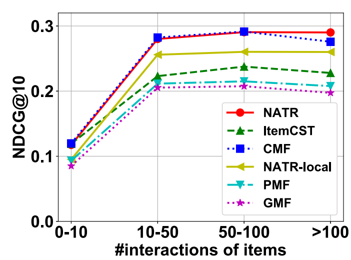
Fig. 4. Training loss and testing performance of NATR, ItemCST, and CMF in each iteration on TC-IQI Dataset.

Section 3.1. The item-based CF is still reserved, even if there is no shared embedding from the auxiliary domain. In other words, those parameters can still be learned on the target domain's data and the item-based CF's power still works. (3) Compared with PMF, GMF, and NATR-local, the proposed NATR outperforms the best of them by 28.26% and 39.11% in HR@10 and NDCG@10 for ML-NF dataset and by 0.23% and 11.09% in HR@10 and NDCG@10 for TC-IQI dataset. The recently-proposed NGCF model achieves similar performance with NATR-local on most metrics, while slightly outperforming NATR-local on some metrics. Nevertheless, our NATR can still steadily outperform NGCF, which demonstrates NATR's effectiveness. We conduct the one-sampled paired t-tests, and we always have p-value<0.05, which demonstrates the performance improvement is stable. It demonstrates that leveraging the item embeddings from the auxiliary domain enhances the recommendation quality in the target domain, which further indicates that the proposed NATR is a promising solution for the cross-domain recommendation task.

— **NATR performs even better than those cross-domain methods with the risk of leaking user privacy.** We can observe that our proposed NATR model achieves the best performance compared to CMF and ItemCST regarding every evaluation metrics in the NF-ML dataset. For HR@10 and NDCG@10, NATR outperforms the best of them by 18.94% and 4.94%, respectively. We guess the reason for such results is that a joint training with data from two domains (CMF and ItemCST) might distract the loss during the optimization and converge at a status which balances the two domains rather than the optimal status of the target domain. We leave further investigations at the future work. To further study how these methods perform, we present the training loss and testing performance in each interaction in Figure 3 (for ML-NF dataset) and Figure 4 (for TC-IQI dataset). For every method in the two figures, we report the best parameter settings. For both datasets, all methods achieve stable performance after about 50 iterations. With fine hyper-parameter



(a) HR@10



(b) NDCG@10

Fig. 5. Performance of all methods on items with different number of interaction records on ML-NF dataset.

tuning to solve over-fitting, our proposed NATR can effectively outperform ItemCST and achieve similar or even better performance than CMF.

— **NATR effectively distills the CF signal encoded in transferred item embeddings.** NATR-local, a degenerative model of our proposed NATR, only utilizes interaction data from the target domain without exploiting transferred item embddings. Specifically, it replaces the transferred item embedding in NATR with local item embeddings. On the one hand, the experimental results in Table 2 show that NATR-local achieves better performance than GMF, a competitive method for single domain, demonstrating that taking the explicit preferences of users into consideration can improve recommendation performance. On the other hand, NATR outperforms NATR-local on two datasets, which means the combining transferred embeddings are better than only a local CF solution on the target domain. This confirms the utility of transferred item embeddings in encoding CF signal from the auxiliary domain.

To summarize, these comparisons on two real-world datasets verify that our proposed NATR model can effectively leverage transferred item embeddings to improve the recommendation performance in the target domain.

4.3 Data Sparsity Problem (RQ2)

As mentioned in the introduction, one of the primary purposes for cross-domain recommendation is to alleviate item data sparsity problem (*i.e.*, items' records are too few) in the target domain. In particular, for those items with few interactions, of which the embeddings cannot be learned well in the target domain itself, transferred embeddings from the auxiliary domain play a bigger role. To study extensively how our proposed NATR model effectively helps to alleviate the item data sparsity issue, we compare the recommendation performance for items with different levels of sparsity.

Specifically, we divide the items into several groups according to the number of interaction records in the training set. Note that each group has a similar number of items, which makes the experimental results more reasonable. Then we apply the evaluation protocol, *leave-one-out*, which is the same as the above experiments. For each item, its performance is defined as the average of HR@10 and NDCG@10 when it is in the test set. We compare the proposed NATR model with all five baseline methods in Figure 5. From the results, we can observe that when the interaction records of an item become sparser, the recommendation performance will go worse. For example, in the first group, of which each item has been interacted by only 1–10 users, the best performance of those single-domain methods is about only 0.220 for HR@10 and 0.098 for NDCG@10. Fortunately, with the help of the auxiliary domain, cross-domain methods can achieve better performance for those sparse items. Out of these methods, our proposed NATR model can

Table 3. Impact of Removing Attention Mechanism

Dataset	ML-NF		TC-IQI	
Methods	**HR@10**	**NDCG@10**	**HR@10**	**NDCG@10**
NATR	0.5781	0.2726	0.6035	0.3365
w/o Item-level Attention	0.5624	0.2655	0.5894	0.3146
w/o Domain-level Attention	0.5669	0.2722	0.5827	0.3204

achieve similar performance compared with CMF and better than ItemCST, which verifies that NATR can serve as a competitive cross-domain recommendation method without sharing user-relevant data.

In summary, our NATR model can improve recommendation performance effectively, no matter the historical records of items are sparse or dense. For items with sparser records, the improvement is more evident and meaningful.

4.4 Impact of Removing Attention Mechanism (RQ3)

In NATR, we utilize the attention mechanism to solve two primary challenges. First, a domain-level attention unit is applied to distill useful signals from transferred item embeddings and integrate them into the target domain. Second, an item-level attention unit is adopted to handle the varying importance of transferred item embeddings. An intuitive question is whether the designed attention unit can really help in our model?

To answer it, we conduct experiments on two degenerative methods of NATR, in which two utilized attention network components are replaced by the simple operation of pooling (*i.e.*, average summation), respectively. We adopt the same evaluation methods with the above experiments, and the performance comparison on two datasets is shown in Table 3. We can observe that removing either item-level attention or domain-level attention will make the recommendation performance worse. Here we have also conducted the one-sampled paired t-tests and we always have the p-value <0.05. This means the original NATR can steadily outperform the de-generated version with one attention unit removed.

To conclude, the experimental results demonstrate the necessity of our two specially designed attention units.

4.5 Impact of Data Sparsity Level in Auxiliary Domain (RQ4)

In our proposed framework of cross-domain recommendation without sharing user-relevant data, the auxiliary domain plays an important role in providing latent embeddings for *bridge items* of two domains. On the other hand, to some extent, the data sparsity of the auxiliary domain affects the quality of those embedding vectors. To study whether our proposed NATR model still outperforms CMF and ItemCST when the quality of embedding vectors is low, we conduct an experiment with different data sparsity levels in the auxiliary domain.

We sample interaction data uniformly with sampling rate 40%, 60%, and 80% while keeping the size of user set and item set unchanged. Similar with experiments in Section 4.2, we obtain item embedding matrix through PMF with fine tuned parameters for these sampled datasets. Then, we apply our proposed NATR model to perform a top-10 recommendation task in the target domain. Note that for ItemCST and CMF, we also conduct experiments with sampled dataset in the auxiliary domain, in a similar way with above experiments. We compare their performance in Table 4. We also present the top-10 recommendation performance of PMF, when generating item embeddings

Table 4. Impact of Data Quality in the Auxiliary Domain

Auxiliary(ML) Data			Methods	Target(NF)	
Sampling Ratio	**HR@10**	**NDCG@10**		**HR@10**	**NDCG@10**
40%	0.6603	0.3410	**NATR**	**0.5086**	**0.2485**
			CMF	0.4508	0.1944
			ItemCST	0.4326	0.1963
60%	0.7452	0.3906	**NATR**	**0.5248**	**0.2666**
			CMF	0.4496	0.1939
			ItemCST	0.4580	0.2094
80%	0.8428	0.4637	**NATR**	**0.5689**	**0.2714**
			CMF	0.4505	0.1945
			ItemCST	0.4637	0.2116

in the auxiliary domain, which represents the quality of the auxiliary data to some extent. From the experimental results, we can have the following observations.

First, our proposed NATR model still outperforms the other two baseline methods, even when auxiliary data is sparse. For HR@10, our proposed NATR model outperforms the best baseline methods by 12.82%, 14.59%, and 22.69% for data ratio 40%, 60%, and 80%, respectively. According to the statistics in Table 1, when only keeping 40% of the interaction data of the auxiliary domain, both two domains' data is sparse. It demonstrates that even the data quality is relatively low in the auxiliary domain, NATR still can effectively extract useful information from the transferred item embeddings and help to improve the recommendation performance in the target domain. Second, we can observe for two feature-transferring based methods, ItemCST and NATR, the change of performance is consistent with the trend of quality change of the auxiliary domain's data. Specifically, a higher-quality item embedding can better improve recommendation in the target domain. This verifies the primary motivation of this article, that item embeddings can encode certain CF signals. On the contrary, CMF achieves a relatively steady performance.

To summarize, our proposed NATR solution still achieves a good enough performance when the data of the auxiliary domain is sparse.

4.6 Impact of Non-Overlapped Items (RQ5)

In this article, we studied the problem of cross-domain recommendation under a widely accepted paradigm that two domains share some overlapped items that serve as the bridge to transfer information across two domains. In real-world applications, not all items are shared by multiple domains operated by different companies. In the aforementioned experiments, to clearly compare our NATR with baseline methods, all items in the dataset are shared by two domains. Therefore, it is essential to study whether our NATR still works or not when there are items only existing at the target domain. In this section, to answer the question, we use the NF-ML dataset with further constraints. To be specific, different from the aforementioned experiments that transfer all items' embeddings, we set a ratio of non-overlapped items. In other words, if the ratio is set to 10%, then 90% of items in the target domain also exist in the auxiliary domain, and these items' play the role of the bridge. We conduct experiments on NF-ML dataset and present the top-K recommendation performance of different ratio with a range of {5%, 10%, 20%, 40%}, in Table 5. From the results, we can observe that with more non-overlapped items, the recommendation performance of NATR on all metrics decrease slowly.

Table 5. Top-K Recommendation Performance on ML-NF Dataset with Different Ratio of Non-overlapped Items (K is set to 1, 2, 5, and 10)

Ratio of Non-overlapped Items	HR@1(NDCG@1)	HR@2	NDCG@2	HR@5	NDCG@5	HR@10	NDCG@10
5%	0.1302	0.1955	0.1390	0.3733	0.2089	0.5721	0.2701
10%	0.1271	0.1917	0.1357	0.3645	0.2056	0.5646	0.2644
20%	0.1224	0.1852	0.1341	0.3573	0.1978	0.5357	0.2569
40%	0.1193	0.1841	0.1292	0.3515	0.1942	0.5273	0.2493

Table 6. Top-K Recommendation Performance Comparison of different Design of Attention Module (K is set to 1, 2, 5, and 10)

	NATR-DotAttn		**NATR-ConcatAttn**	
K	**HR@K**	**NDCG@K**	**HR@K**	**NDCG@K**
1	0.1315	0.1315	0.1280	0.1280
2	0.1976	0.1403	0.1965	0.1389
5	0.3776	0.2110	0.3771	0.2096
10	0.5781	0.2726	0.5715	0.2691

4.7 Impact of Design of Attention Module (RQ6)

In our proposed NATR, we adopt two-level attention mechanisms, and in both, there is an element-wise product operation to combine two parts of embeddings. In fact, we have other choices, such as the concatenation operation. In this section, we conduct experiments to study the recommendation performance if we replace the element-wise product with concatenation. Here we name it NATR-ConcatAttn and present the recommendation performance in Table 6. We can observe that replacing the element-wise product with concatenation causes a very small drop in the performance of top-K recommendation. This verifies that both two variants of attention are effective. Besides, this small drop can be explained that the corresponding relation of each dimension in latent space is lost when we use concatenation operation. Nevertheless, the strong power of neural networks can still effectively learn the interaction of two vectors, making the performance still good enough.

In conclusion, extensive experiments on two real-world datasets verify the efficacy of our proposed model, and further studies demonstrate that our model can alleviate the data sparsity problem and achieve good performance when there are non-overlapped items. Moreover, the utility of our specially designed attention network components is verified.

5 RELATED WORK

In this article, we propose a solution for cross-domain recommendation without sharing user-relevant data based on neural networks. The closed related work can be divided into cross-domain recommendation and neural network based recommendation.

Cross-domain Recommendation. To alleviate cold start and data sparsity issue, cross-domain recommendation is a typical solution which takes data from multiple domains into consideration [31]. With the help of the auxiliary domain, cross-domain recommendation methods can achieve better performance (*i.e.* recommendation accuracy) than single-domain ones. Approaches of cross-domain recommendation can be broadly classified into two types: collaborative and content-based.

Collaborative cross-domain recommendation refers to those approaches utilizing interaction data (rating history, for example) from two domains. Ajit et al. [3] proposed an MF-based model, CMF, which assumes a common global user factor matrix for all domains, and it factorizes matrices from multiple domains simultaneously. Li et al. [32] proposed a model named Code Book Transfer, which builds a matrix named codebook to represent cluster-level rating pattern, and this codebook is shared by two domains. A recent study [2] considered a special task in which both users and items are overlapped, and they proposed an MF-based model that assumes part of the user embeddings and whole item embeddings are shared across domains. With a similar setting, Man et al. [33] proposed a neural method that employs multi-layer perceptron to adapt user and item embeddings between two domains. Pan et al. [10] utilize auxiliary interaction data with a regularization term concerned with overlapped user and item in objective function in the MF model. Do et al. [34] discovered both explicit and implicit similarities from latent factors across domains based on CMF. Another category of cross-domain recommendation models is content-based ones, which sharing attributes of user or items from auxiliary domain [1, 35, 36]. Agarwal et al. [1] proposed an MF-based model in cross-domain recommendation when multi-modal user profiles are available. Elkahky et al. [35] transformed user profile and item attributes to dense vectors through deep neural networks and matched them in latent space. Zhang et al. [36] utilize textual, structure, and visual knowledge of items as the auxiliary domain to aid in building item embedding.

In this article, we focus on collaborative cross-domain recommendation with auxiliary interaction data, a widely used setting in literature. Specifically, our problem is a typical system-level cross-domain recommendation task, where the same items are shared across domains, according to the definition in two surveys [37, 38]. There is a common assumption in existing methods that the whole interaction data can be fully shared across domains, which has the risk of leaking user privacy since various domains may be operated by different companies. A very close related work is [39], which studies protecting user privacy in the task of cross-domain location recommendation. However, this work is only applicable to location data, and it still transfers user-relevant data, even if it has been perturbated by the protection mechanism. In this work, we advocate a more realistic setting that only item-side data can be shared.

Neural Network Based Recommendation. Salakhutdinov et al. [40] proposed Restricted Boltzmann Machines to predict explicit ratings, which was the first work to apply neural networks to recommender systems. Recently, similar to the research field of CV and NLP, neural networks have achieved great success in recommender systems. In general, neural network-based recommendation can be divided into two categories. The first category of researches proposes to utilize neural networks to learn the matching function between users and items. He et al. proposed a model named neural collaborative filtering, which utilizes neural networks to replace the inner product in MF. This is further extended [8, 41, 42] to more complicated neural interaction functions. Tay et al. [29] approached the neural interaction function from another perspective and proposed a relational-translation based neural match function. Sedhain et al. [43] first proposed to use autoencoder to extract CF signal, and recently it is extended by [44] with variational autoencoder. Recently, inspired by the advances in **graph convolutional network (GCN)**, some works [45, 46] proposed to model user interaction with graph and proposed GCN-based recommendation methods.

The second category of neural-based recommendation methods takes advantage of neural networks' strong power of extracting latent representation from complicated and complex data. There are a series of neural-based extensions, such as DeepFM [47], AFM [48], and xDeepFM [49] of factorization machine [50], that adopt neural network to capture feature interaction in the content-based recommendation. Besides, some works utilized neural networks to extract certain auxiliary data, such as social network [51], textual data [52, 53], fashion data [54], knowledge graph [55], sequential behavior [56, 57], and activity data [58].

In this article, we propose a novel neural model to solve the challenges of extracting useful knowledge from item embedding of the auxiliary domain with the power of neural networks in representative learning.

Privacy-preserving Recommendation. As one of the hottest personalized service in today's online systems, recommendation is close to users' personal data such as demographics or behavioral data. Current researches have shown that there is a high risk of leaking user privacy in traditional recommender system since even though the collected preferences such as movie ratings are not so sensitive, these can be utilized to infer sensitive user information [59, 60]. This concern pushes forward the development of privacy-preserving recommendation. Some early approaches [6, 59, 61–63] assume that recommendation engine itself can be trusted and protect user privacy from the attack of the third party. With such assumption, recommenders can collect users' raw data; and then a protection mechanism is applied to the learned recommendation model or generated recommendation results as the third party may infer user interests with the released model or results. However, such an assumption is not so reasonable as the recommender may be not reliable. Therefore, recent researches [64–69] pay more attention to propose privacy-preserving methods for recommendation without trusting anyone. A major category of approaches [64–66] rely on distributed MF to train an MF model in a decentralized manner, avoiding users uploading raw data to the server. With exchanging gradient or parameter, the MF model can successfully converge, fit data, and generate recommendation results. Another category of approaches for distrusted recommender is to apply data protection mechanism during data collection [6, 67, 68]. With the mechanism, some noise is added to the raw data, and the server can only access noisy data. Differential privacy [23], which is a concept from the database community, is frequently used to provide privacy guarantee in these researches [6, 67, 70]. Following our previous work [11], in this article, we approach the problem of protecting user privacy in the task of cross-domain recommendation.

Recently, Chen et al. [71] propose to introduce federated learning into recommendation. Federated learning is a kind of machine learning under a special setting that multiple federations jointly participate in the learning procedure without sharing raw data. However, it is required in federated learning based recommendation that there is adequate context to make sure knowledge about predictive signal can be shared or transferred across federations.

Different from existing methods, we propose to transfer user-irrelevant data, embedding of bridge items, to share predict signal and improve recommendation performance.

6 CONCLUSION

In this work, we present a new cross-domain recommendation solution, which can avoid user privacy leakage by transferring only item embeddings from the auxiliary domain. To better exploit the transferred item embeddings, we propose a neural network method named NATR, combining item-level and domain-level attention mechanisms to address the challenges in cross-domain learning. We conduct extensive experiments on two real-world datasets, demonstrating that our NATR method can improve the recommendation performance of the target domain by 18.94%. To the best of our knowledge, this is the first work that concerns user privacy in cross-domain recommendation, and presents a sound solution to exploit the predictive signal without sharing any user-relevant information.

There are three points about this work that we plan to address in future. First, although our method has taken the dimensionality discrepancy into consideration via a dimension-adaption fully-connected layer, we only empirically verify its effectiveness when two domains have the same embedding size. As such, we will study how different sizes of transferred embeddings would affect the recommendation performance. Second, we will study how much private information, including user interaction history and user profiles, a potential attacker can obtain if we share user embeddings across two domains. This can provide more formal privacy analysis and further support our motivation of avoiding sharing user embedding vectors.

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, and Bo Long. 2011. Localized factor models for multi-context recommendation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. 609–617.

[2] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. 2017. Multi-site user behavior modeling and its application in video recommendation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17)*. 175–184.

[3] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. 650–658.

[4] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *International World Wide Web Conference (WWW'13)*. 595–606.

[5] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-domain collaboration recommendation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. 1285–1293.

[6] Chen Gao, Chao Huang, Yue Yu, Huandong Wang, Yong Li, and Depeng Jin. 2019. Privacy-preserving cross-domain location recommendation. In *ACM International Conference on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT'19)*.

[7] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored item similarity models for top-n recommender systems. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. 659–667.

[8] Xiangnan He, Zhenkui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.

[9] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems* 22, 1 (2004), 143–177.

[10] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *The AAAI Conference on Artificial Intelligence (AAAI'10)*, Vol. 10. 230–235.

[11] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. 2019. Cross-domain recommendation without sharing user-relevant data. In *The World Wide Web Conference (WWW'19)*. 491–502.

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

[13] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for YouTube recommendations. In *ACM Conference on Recommender Systems (RecSys'16)*. 191–198.

[14] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. In *Deep Learning for Recommender Systems Workshop (DLRS'16)*. 7–10.

[15] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *The Conference on Uncertainty in Artificial Intelligence (UAI'09)*. 452–461.

[16] Andriy Mnih and Ruslan R. Salakhutdinov. 2008. Probabilistic matrix factorization. In *Annual Conference on Neural Information Processing Systems (NIPS'08)*. 1257–1264.

[17] Jun Xu, Xiangnan He, and Hang Li. 2018. Deep learning for matching in search and recommendation. *International World Wide Web Conference (WWW'18)*.

[18] 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 *International Joint Conferences on Artificial Intelligence (IJCAI'17)*. 3119–3125.

[19] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive recurrent social recommendation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18)*. 185–194.

[20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *International World Wide Web Conference (WWW'17)*. 173–182.

[21] Christian Robert. 2014. Machine learning, a probabilistic perspective. *CHANCE* 27, 2 (2014), 62–63.

[22] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems* 38, 2 (2020), 1–28.

[23] Cynthia Dwork. 2011. Differential privacy. In *Proceedings of the International Conference on Theory And Applications of Models of Computation*. Springer, 1–19.

[24] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.

[25] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 106–115.

[26] Chanyoung Park, Donghyun Kim, Jinoh Oh, and Hwanjo Yu. 2017. Do also-viewed products help user rating prediction?. In *International World Wide Web Conference (WWW'17)*. 1113–1122.

[27] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *International Conference on Data Mining (ICDM'08)*. 263–272.

[28] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. 426–434.

[29] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *International World Wide Web Conference (WWW'18)*.

[30] Ting Bai, Ji-Rong Wen, Jun Zhang, and Wayne Xin Zhao. 2017. A neural collaborative filtering model with interaction-based neighborhood. In *International Conference on Information and Knowledge Management (CIKM'17)*. 1979–1982.

[31] Nima Mirbakhsh and Charles X. Ling. 2015. Improving top-n recommendation for cold-start users via cross-domain information. *ACM Transactions on Knowledge Discovery from Data* 9, 4 (2015), 1–19.

[32] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. In *International Joint Conferences on Artificial Intelligence (IJCAI'09)*, Vol. 9. 2052–2057.

[33] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach. In *International Joint Conferences on Artificial Intelligence (IJCAI'17)*. 2464–2470.

[34] Quan Do, Wei Liu, Jin Fan, and Dacheng Tao. 2021. Unveiling hidden implicit similarities for cross-domain recommendation. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2021), 302–315.

[35] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *International World Wide Web Conference (WWW'15)*. 278–288.

[36] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*.

[37] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. In *Recommender Systems Handbook*. Springer, 919–959.

[38] Bin Li. 2011. Cross-domain collaborative filtering: A brief survey. In *International Conference on Tools with Artificial Intelligence*. 1085–1086.

[39] Chen Gao, Xiangnan He, Danhua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, Lina Yao, Yang Song, and Depeng Jin. 2021. Learning to recommend with multiple cascading behaviors. *IEEE Transactions on Knowledge and Data Engineering* 33, 6 (2021), 2588–2601. DOI: 10.1109/TKDE.2019.2958808

[40] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *International Conference on Machine Learning (ICML'07)*. 791–798.

[41] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. In *International Joint Conferences on Artificial Intelligence (IJCAI'18)*.

[42] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems. In *International Joint Conferences on Artificial Intelligence (IJCAI'17)*. 3203–3209.

[43] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders meet collaborative filtering. In *International World Wide Web Conference (WWW'15)*. ACM, 111–112.

[44] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *International World Wide Web Conference (WWW'18)*. ACM, 689–698.

[45] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18)*.

[46] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[47] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *International Joint Conferences on Artificial Intelligence (IJCAI'17)*. 1725–1731.

[48] 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 *International Joint Conferences on Artificial Intelligence (IJCAI'17)*.
[49] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18)*.
[50] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining (ICDM'10)*. 995–1000.
[51] Guang-Neng Hu, Xin-Yu Dai, Feng-Yu Qiu, Rui Xia, Tao Li, Shu-Jian Huang, and Jia-Jun Chen. 2018. Collaborative filtering with topic and social latent factors incorporating implicit feedback. *ACM Transactions on Knowledge Discovery from Data* 12, 2 (2018), 1–30.
[52] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *International World Wide Web Conference (WWW'18)*. ACM, 1835–1844.
[53] Jaegul Choo, Hannah Kim, Edward Clarkson, Zhicheng Liu, Changhyun Lee, Fuxin Li, Hanseung Lee, Ramakrishnan Kannan, Charles D. Stolper, John Stasko, and Haesun Park. 2018. VisIRR: A visual analytics system for information retrieval and recommendation for large-scale document data. *ACM Transactions on Knowledge Discovery from Data* 12, 1 (2018), 1–20.
[54] Ruiping Yin, Kan Li, Jie Lu, and Guangquan Zhang. 2019. Enhancing fashion recommendation with visual compatibility relationship. In *International World Wide Web Conference (WWW'19)*. ACM, 3434–3440.
[55] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *International World Wide Web Conference (WWW'19)*. ACM, 2000–2010.
[56] Hung-Hsuan Chen. 2018. Behavior2Vec: Generating distributed representations of user's behaviors on products for recommender systems. *ACM Transactions on Knowledge Discovery from Data* 12, 4 (2018), 1–20.
[57] Yu Lei and Wenjie Li. 2019. Interactive recommendation with user-specific deep reinforcement learning. *ACM Transactions on Knowledge Discovery from Data* 13, 6 (2019), 1–15.
[58] Jianmo Ni, Larry Muhlstein, and Julian McAuley. 2019. Modeling heart rate and activity data for personalized fitness recommendation. In *The World Wide Web Conference*. ACM, 1343–1353.
[59] Frank McSherry and Ilya Mironov. 2009. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. 627–636.
[60] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. 2011. "You might also like:" Privacy risks of collaborative filtering. In *IEEE Symposium on Security and Privacy (S&P'11)*. 231–246.
[61] Peilin Zhao Jun Zhou Xiaolong Li Chaochao Chen, Ziqi Liu. 2018. Privacy preserving point-of-interest recommendation using decentralized matrix factorization. In *The AAAI Conference on Artificial Intelligence (AAAI'18)*.
[62] Daniele Riboni and Claudio Bettini. Private context-aware recommendation of points of interest: An initial investigation. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. 584–589.
[63] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying differential privacy to matrix factorization. In *9th ACM Conference on Recommender Systems (RecSys'15)*. 107–114.
[64] Qing Ling, Yangyang Xu, Wotao Yin, and Zaiwen Wen. 2012. Decentralized low-rank matrix completion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'12)*. 2925–2928.
[65] István Hegedűs, Árpád Berta, Levente Kocsis, András A Benczúr, and Márk Jelasity. 2016. Robust decentralized low-rank matrix decomposition. *ACM Transactions on Intelligent Systems and Technology* 7, 4 (2016), 62.
[66] Yu Xin and Tommi Jaakkola. 2014. Controlling privacy in recommender systems. In *Advances in Neural Information Processing Systems (NeurIPS'14)*. 2618–2626.
[67] Chao Li, Balaji Palanisamy, and James Joshi. 2017. Differentially private trajectory analysis for points-of-interest recommendation. In *2017 IEEE International Congress on Big Data (BigData Congress'17)*. IEEE, 49–56.
[68] Huseyin Polat and Wenliang Du. 2005. SVD-based collaborative filtering with privacy. In *ACM Symposium on Applied Computing*. 791–795.
[69] Beyza Ermiš and A. Taylan Cemgil. 2020. Data sharing via differentially private coupled matrix factorization. *ACM Transactions on Knowledge Discovery from Data* 14, 3 (2020), 1–27.
[70] Chen Gao, Chao Huang, Dongsheng Lin, Depeng Jin, and Yong Li. 2020. DPLCF: Differentially private local collaborative filtering. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 961–970.
[71] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876* (2018).

Received June 2020; revised December 2020; accepted January 2021