HERS: Modeling Influential Contexts with Heterogeneous Relations for Sparse and Cold-Start Recommendation

Liang Hu,[1,2] Songlei Jian,[1,3] Longbing Cao,[1] Zhiping Gu,[4] Qingkui Chen,[2] Artak Amirbekyan[5]
[1]Advanced Analytics Institute, University of Technology Sydney, Australia
[2]University of Shanghai for Science and Technology, China, [3]National University of Defense Technology, China
[4]Shanghai Technical Institute of Electronics Information, China, [5]Commonwealth Bank of Australia
rainmilk@gmail.com, jiansonglei@163.com, longbing.cao@uts.edu.au
guzhiping@stiei.edu.cn, chenqingkui@usst.edu.cn, artak.amirbekyan@cba.com.au

Abstract

Classic recommender systems face challenges in addressing the data sparsity and cold-start problems with only modeling the user-item relation. An essential direction is to incorporate and understand the additional heterogeneous relations, e.g., user-user and item-item relations, since each user-item interaction is often influenced by other users and items, which form the user's/item's influential contexts. This induces important yet challenging issues, including modeling heterogeneous relations, interactions, and the strength of the influence from users/items in the influential contexts. To this end, we design Influential-Context Aggregation Units (ICAU) to aggregate the user-user/item-item relations within a given context as the influential context embeddings. Accordingly, we propose a Heterogeneous relations-Embedded Recommender System (HERS) based on ICAUs to model and interpret the underlying motivation of user-item interactions by considering user-user and item-item influences. The experiments on two real-world datasets show the highly improved recommendation quality made by HERS and its superiority in handling the cold-start problem. In addition, we demonstrate the interpretability of modeling influential contexts in explaining the recommendation results.

Introduction

Recommender systems (RSs) are essentially embedded with heterogeneous relations: user-user couplings, item-item couplings, and user-item couplings (Cao 2015), with which we call Heterogeneous relations Embedded Recommender Systems (HERS). It is increasingly recognized that modeling such multiple heterogeneous relations is essential for understanding the non-IID nature and characteristics of RSs (Cao 2016) and addressing such challenges as social, group-based, context-based, cross-domain, sparse, cold-start, dynamic and deep recommendation (Tang, Hu, and Liu 2013; Hu et al. 2014; 2017b; 2016; Do and Cao 2018a; 2018b; Zhang et al. 2018).

Figure 1 illustrates the concept and motivation of a HERS containing user-user relations (e.g., friendship), item-item relations (e.g., the same category), and user-item relations (e.g., users' purchase or feedback on items). Given a user u_t in the user social network, her/his selections are often influenced by friends (Friedkin and Johnsen 2011). Moreover,

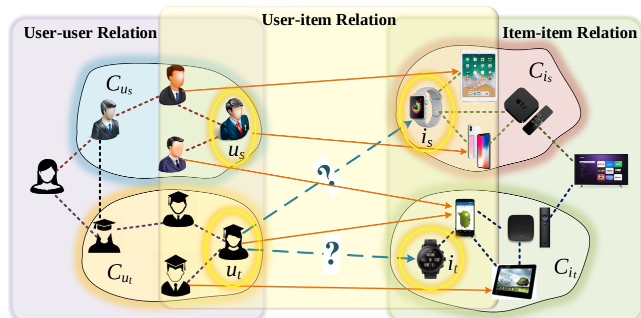


Figure 1: An HERS consists of three heterogeneous relations: user-user, item-item, and user-item. Each user's choice is relevant to the corresponding user's and item's influential contexts.

the influences from friends' friends may transitively influence u_t's choice. C_{u_t} signifies the influential context w.r.t. u_t, which outlines u_t's influential neighbors and the relationships between them. Those users in C_{u_t} have the considerable influence on u_t' selection. Similarly, user selection on an item i_t is also influenced by i_t's relevant items which form i_t's influential context C_{i_t}. For example, we can infer that u_t more probably selects i_t (Android watch) than i_s (Apple watch) if we consider the corresponding influential contexts C_{u_t}, C_{i_t} and C_{i_s}, i.e., the choices from u_t's friends and the compatibility of electronic products. This example shows that the influential contexts of users and items indicate how a user's choice on items is made, thus making recommendation more accurate and interpretable. The viewpoint of HERSs and influential contexts can further address the sparsity of user-item interactions and both user and item cold-start problems by referring to the influential users and items in the contexts.

Modeling an HERS with influential contexts involves various challenges. In this work, we focus on two major ones: (1) *How to model a user's and an item's influential contexts behind an observed user-item interaction?* (2) *How to learn the strength of influence from different users or items in the different contexts?* These two questions are important and challenging because the influential contexts for different user-item interactions are different, and the influence from each user/item in one context is also different.

Some relevant work includes social RSs (Tang, Hu, and Liu 2013; Jiang et al. 2014; Hu et al. 2017a) which combine collaborative filtering (CF) and social network analysis (Vasuki et al. 2010; Tang et al. 2012) by co-factorizing the rating matrix and the social-relation matrix (Ma et al. 2008; Jamali and Ester 2010) and adding regularization term according to the social closeness (Ma et al. 2011) in matrix factorization (MF). However, they only consider user relationships but ignore the influence of relevant items and only model first-order influences from neighbors and fail to consider higher-order influences from indirect users. Factorization machine (FM) (Rendle 2012) represents multiple relations with a design matrix (Rendle 2013). However, using a single design matrix for all relations implicitly assumes the homogeneity of these relations. FM is unable to model the strengths of influence from the same users/items in different influential contexts w.r.t. different target users/items. Recent work on non-IID Rs includes modeling user/item couplings in explicit user/item metadata and latent user/item factors by coupled MF (Li, Xu, and Cao 2015), coupled Poisson factorization (Do and Cao 2018a) and deep models CoupledCF (Zhang et al. 2018).

In this paper, we design Influential-Context Aggregation Units (ICAUs) to aggregate all user/item influences in a context into an embedding, namely Influential Context Embedding (ICE). Taking ICAUs as the building blocks, we construct an HERS that considers both user's and item's influential contexts when making recommendations. The main contributions of this work include:

- An HERS framework is proposed to model the heterogeneous relations in recommendation tasks by considering both the user's and the item's influential contexts.
- As the building block of the HERS framework, ICAUs are designed to aggregate all users'/items' influences in a context into an ICE as the representation of this influential context.
- The ICAU-based HERS empowers the interpretability on recommended results in terms of measuring the strength of influence from relevant users and items.

We conduct extensive experiments on two real datasets with heterogeneous relations. The results show the effectiveness of our method on recommendation quality and its superiority in terms of handling both user and item cold-start and sparsity problems. In addition, we visualize the learned influential contexts to interpret the recommendations.

Related Work

We mainly review the work which somehow incorporate the user-user, user-item and item-item coupling relationships in RS (Cao 2016). With regard to the user-user relation, classic RSs consider social relationships in RS by extending CF models such as by modeling social network relations (Tang, Hu, and Liu 2013), SoRec (Ma et al. 2008) to co-factorize the user-item matrix and the user-user social relation matrix by sharing the common user latent factor matrix, and SocialMF (Jamali and Ester 2010) and SoReg (Ma et al. 2011) to regularize the user latent factor vector of a target user via the user latent factor vectors of their trusters. However, social RSs only incorporate additional information from user side but ignoring the relevance between items.

Several recent works jointly model multiple relations. Collective Matrix Factorization (CMF) (Singh and Gordon 2008) represents these relations in terms of a collection of coupled matrices with sharing the latent factors along the coupled dimensions. Then, CMF conducts co-factorization on these coupled matrices to learn the latent factor matrices. Factorization Machine (FM) (Rendle 2012) factorizes the interactions between each pair of features via a design matrix to represent multiple relations (Rendle 2013). However, the above models assume the relations are homogeneous but ignore the heterogeneous interactions within and between users and items. These factorization methods only model the first-order influences from direct neighbors and fail to consider to higher order influences in a context.

More recent work is to model both explicit and implicit user-user, item-item, and user-item couplings (Cao 2015; 2016). Coupled Matrix Factorization (Li, Xu, and Cao 2015) involves user couplings and item couplings into MF. Coupled Poisson Factorization (Do and Cao 2018a) integrates user/item metadata and their relations into Poisson Factorization for large and sparse recommendation. Dynamic Matrix Factorization mGDMF (Do and Cao 2018b) involves a conjugate Gamma-Poisson model by incorporating metadata influence to effectively and efficiently model massive, sparse and dynamic recommendations. Deep neural models have also been incorporated into RSs, e.g., (Wang et al. 2017) extended the Neural Collaborative Filtering (NCF) (He et al. 2017) to cross-domain social recommendations, and a Neural Social Collaborative Ranking (NSCR) model based on NCF (Wang et al. 2017) seamlessly integrates user-item interactions from the information domain and the user-user relationships from the social domain. The Neural Factorization Machine (NFM) extends FM with neural networks by adding multiple hidden layers to learn non-linear interactions. CoupledCF (Zhang et al. 2018) learns explicit and implicit user-item couplings in recommendation for deep collaborative filtering. However, all these methods only model pairwise interactions instead of all influences in the influential contexts. Moreover, they cannot tell the strengths of influence from each user or item.

The Neural Factorization Machine (NFM) extends FM with neural networks by adding multiple hidden layers to learn non-linear interactions. However, all these methods only model pairwise interactions instead of all influences in the influential contexts. Moreover, they cannot tell the strengths of influence from each user or item.

Problem Formulation and Model Architecture

In this section, we formulate the influences of multiple heterogeneous relations in RSs, and then present the framework of modeling influential user/item contexts.

Problem Formulation

In this paper, we aim to build an HERS which exploits the information learned from user-user, user-item and user-item

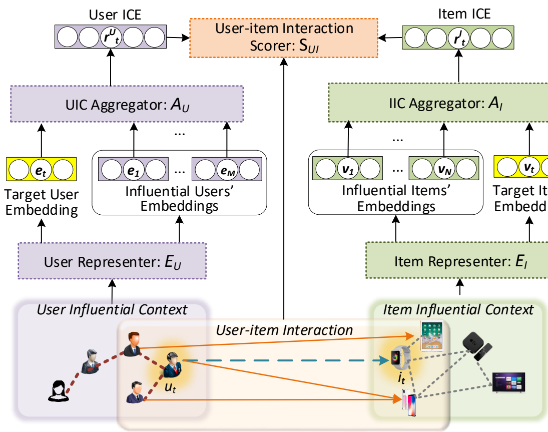


Figure 2: The architecture of HERS for modeling user-item interaction with user's and item's influential contexts

relations for more effective recommendation. The architecture of proposed HERS model is illustrated in Figure 2.

Let u and $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$ denote a user and the whole user set. $\mathcal{R}_{\mathcal{U}}$ denotes the user-user relation. Let i and $\mathcal{I} = \{i_1, i_2, \cdots, i_{|\mathcal{I}|}\}$ denote an item and the whole item set. $\mathcal{R}_{\mathcal{I}}$ denotes the item-item relation. The user-item interactions are denoted as $\mathcal{R}_{\mathcal{UI}} = \{y_{u,i} | u \in \mathcal{U}, i \in \mathcal{I}\}$, and $y_{u,i}$ could be explicit feedback, e.g., ratings or implicit feedback, e.g., clicks. Generally, each user-item interaction $y_{u,i}$ is not only decided by user u and item i but also other users and items in the influential context (Cao 2016). Hence, we formally define a user's and an item's influential contexts to model the user-item interactions.

Definition 1. User Influential Context (UIC): Given a target user u, the UIC denotes $\mathcal{C}_u = \{\mathcal{U}_u, \mathcal{R}_u\}$, where $\mathcal{U}_u = \{u, \mathcal{U}_u^c\}$ consists of target user u and all influential users relevant to u, $\mathcal{R}_u$ denotes the user relationships over $\mathcal{U}_u$.

Definition 2. Item Influential Context (IIC): Given a target item i, the IIC denotes $\mathcal{C}_i = \{\mathcal{I}_i, \mathcal{R}_i\}$, where $\mathcal{I}_i = \{i, \mathcal{I}_i^c\}$ consists of target item i and all influential items relevant to i, $\mathcal{R}_i$ denotes all the item relationships over $\mathcal{I}_i$.

Interaction Score Decomposition: Each user-item interaction $y_{u,i}$ can be measured by a score function s in terms of the UIC $\mathcal{C}_u$ and the IIC $\mathcal{C}_i$; formally, $s : s(\mathcal{C}_u, \mathcal{C}_i, y_{u,i}) \mapsto s_{\langle \mathcal{C}_u, \mathcal{C}_i \rangle}$. According to Definitions 1 and 2, the overall interaction score $s_{\langle \mathcal{C}_u, \mathcal{C}_i \rangle}$ can be decomposed into four scores:

$$s_{\langle \mathcal{C}_u, \mathcal{C}_i \rangle} = \lambda_1 s_{\langle u,i \rangle} + \lambda_2 s_{\langle u, \mathcal{I}_i^c \rangle} + \lambda_3 s_{\langle U_u^c, i \rangle} + \lambda_4 s_{\langle U_u^c, \mathcal{I}_i^c \rangle} \quad (1)$$

where $s_{\langle u,i \rangle}$ scores user u's preference on item i; $s_{\langle u, \mathcal{I}_i^c \rangle}$ scores u's preference on influential items $\mathcal{I}_i^c$; $s_{\langle U_u^c, i \rangle}$ scores relevant users' preference on i, and $s_{\langle U_u^c, \mathcal{I}_i^c \rangle}$ scores the subsidiary preference between influential users and influential items. $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the scale parameters for weighing these scores.

Model Architecture

The architecture of HERS with the user's and item's influential contexts is illustrated in Figure 2, which consists of five components: User Representer E_U, UIC Aggregator A_U, Item Representer E_I, IIC Aggregator A_I and user-item interaction scorer S_{UI}. Given a target user u_t and the corresponding UIC $\mathcal{C}_{u_t}$, a target item i_t and the corresponding IIC $\mathcal{C}_{i_t}$:

- User Representer E_U: it maps target user u_t and its influential users in UIC to the corresponding user embeddings, i.e., $E_U(\mathcal{U}_{u_t}) \mapsto \mathcal{E}_{u_t}$ where $\mathcal{E}_{u_t} = \{\mathbf{e}_t, \mathbf{e}_1, \cdots \mathbf{e}_M\}$.
- Item Representer E_I: it maps target item i_t and its influential items in IIC to the corresponding item embeddings, i.e., $E_I(\mathcal{I}_{i_t}) \mapsto \mathcal{E}_{i_t}$ where $\mathcal{E}_{i_t} = \{\mathbf{v}_t, \mathbf{v}_1, \cdots \mathbf{v}_N\}$.
- UIC Aggregator A_U: it learns a representation $\mathbf{r}_t^U$ for the influential context $\mathcal{C}_{u_t}$, namely influential context embedding (ICE). Formally, we have $A_U(\mathcal{C}_{u_t}, \mathcal{E}_{u_t}) \mapsto \mathbf{r}_t^U$.
- IIC Aggregator A_I: it learns i_t's ICE by aggregating the influential context $\mathcal{C}_{i_t}$, that is, $A_I(\mathcal{C}_{i_t}, \mathcal{E}_{i_t}) \mapsto \mathbf{r}_t^I$.
- User-item Interaction Scorer S_{UI}: it learns to score the interaction strength between the target user-item pair $\langle u_t, i_t \rangle$ in terms of the user ICE $\mathbf{r}_t^U$ and the item ICE $\mathbf{r}_t^I$, namely $S_{UI}(\mathbf{r}_t^U, \mathbf{r}_t^I, y_{u_t,i_t}) \mapsto s_{\langle \mathcal{C}_u, \mathcal{C}_i \rangle}$ (cf. Eq. 1).

This architecture can be implemented w.r.t. many concrete methods, e.g., a mixture model. In this paper, we implement it w.r.t. neural networks which have been proved mostly effective and efficient in recent years.

Neural ICAU-based HERS Model

In this section, we present the details of a neural model to implement the HERS architecture illustrated in Figure 2. Specially, we design the neural ICAUs to learn ICEs as the core component of this HERS model.

Influential-context Aggregation Unit

The ICAUs aim to aggregate the user embeddings $\mathcal{E}_{u_t}$ or the item embeddings $\mathcal{E}_{i_t}$ in a context into an ICE according to the strength of influence from each user or item. Figure 3 demonstrates an ICAU for aggregating user embeddings $\mathcal{E}_{u_t}$, which consists of a two-stage aggregation: S1 and S2.

S1: This stage outputs the subsidiary influence embedding $\mathbf{c}_t$ through an aggregation function $h(\cdot)$ over the influential users' embeddings:

$$\{\alpha_1, \cdots \alpha_K\} = a(\mathbf{e}_1, \cdots \mathbf{e}_K) \quad (2)$$

$$\mathbf{c}_t = h(\mathbf{e}_1, \cdots \mathbf{e}_K | \alpha_1, \cdots \alpha_K). \quad (3)$$

where α_i denotes the influential strength modeled by a function $a(\cdot)$.

S2: This stage generates the ICE by aggregating the subsidiary influence context embedding $\mathbf{c}_t$ and the target embedding $\mathbf{e}_t$ through a gate function $f(\cdot)$:

$$g = f(\mathbf{c}_t, \mathbf{e}_t) \quad (4)$$

$$\mathbf{r}_t = g\mathbf{c}_t + (1 - g)\mathbf{e}_t \quad (5)$$

In ICAU, h and f could be any linear or non-linear functions. In this work, we implement h by multilayer neural networks in terms of the attention mechanism. And f is implemented with a gate neural network. Note that the ICAUs can be used in cascade to aggregate higher order ICEs, which is presented in the next subsection.

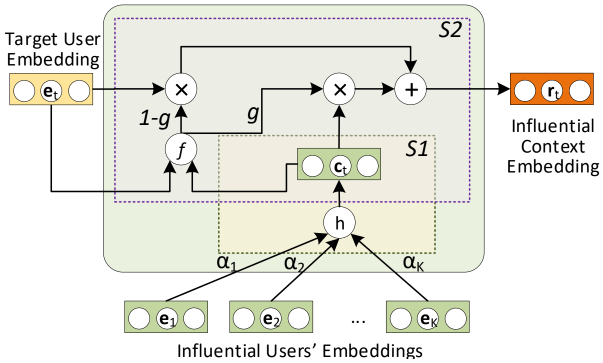


Figure 3: Influential-Context Aggregation Unit (ICAU): A two-stage aggregation model to construct ICE

User's Influential Context Embedding

Given a user influential context $\mathcal{C}_{u_t} = \{\mathcal{U}_{u_t}, \mathcal{R}_{u_t}\}$ (cf. Definition 1), $\mathcal{U}_{u_t}$ consists of the target user u_t and her/his first-order influential neighbors $\{u_{t,m}\}_{1\le m\le M}$, and each $u_{t,m}$'s neighbors $\{u_{t,m,k}\}_{1\le k\le K_m}$, i.e., second-order influential neighbors of u_t, according to the relationships $\mathcal{R}_{u_t}$.

To learn the ICE from the UIC $\mathcal{C}_{u_t}$, we first employ the User Representer E_U to extract the user embeddings for all users in $\mathcal{U}_{u_t}$. The embedding for a first-order neighbor $u_{t,m}$ is denoted as $\mathbf{e}_{t,m}$, and the embedding for a second-order neighbor $u_{t,m,k}$ is denoted as $\mathbf{e}_{t,m,k}$. To generate the ICE for $\mathcal{C}_{u_t}$, we construct an ICAU-based two-level tree-like model as the UIC Aggregator A_U to recursively embed both second-order and first-order neighbors' influences. Firstly, the second level ICAUs are used to generate the ICE $\mathbf{r}_{t,m}$ w.r.t. each first-order neighbor $u_{t,m}$. Then, the first level ICAU recursively generates the target ICE $\mathbf{r}_t^U$ by aggregating these first-order ICEs $\{\mathbf{r}_{t,m}\}$. More details are presented as follows.

The Second Level ICAUs Taking a first-order influential neighbor $u_{t,m}$ of target user u_t as an example, we present how to implement an ICAU with neural networks for generating the ICE $\mathbf{r}_{t,m}$ w.r.t. $u_{t,m}$ in two stages.

S1: We adopt the attention mechanism to model the influential strength for each neighbor $u_{t,m,k}$ of $u_{t,m}$. Specifically, we construct a three-layer attention neural network to weight the influence according to the user embedding $\mathbf{e}_{t,m,k}$. First, $\mathbf{e}_{t,m,k}$ is projected into hidden units by a $\tanh$ layer to capture nonlinear interaction:

$$\mathbf{h}_{t,m,k} = \tanh(\mathbf{W}^{(1)}\mathbf{e}_{t,m,k} + \mathbf{b}) \tag{6}$$

where the weight matrix $\mathbf{W}^{(1)} \in \mathbb{R}^{L\times L}$ and we omit the bias term $\mathbf{b}$ in the following equations for concision.

Then, the normalized influence for each neighbor is scored by the $\mathrm{softmax}(x_k) = e^{x_k}/\sum_j e^{x_j}$ function:

$$\alpha^U_{t,m,k} = \mathrm{softmax}\Big(isr_\theta\big(\mathbf{W}^{(2)}\mathbf{h}_{t,m,k}\big)\Big) \tag{7}$$

where the weight matrix $\mathbf{W}^{(2)} \in \mathbb{R}^{1\times L}$ and isr_θ is an inverse square root unit which is defined as follows:

$$isr_\theta(x) = \frac{x}{\sqrt{1+\theta x^2}} \tag{8}$$

where θ is the parameter which decides the range of function $isr_\theta(\cdot)$. A large θ makes the upper bound and lower bound close to 0; as a result, the softmax tends to output uniform weights. On the contrary, the softmax tends to output a single large weight with a small θ.

Further, the subsidiary influence embedding $\mathbf{c}_{t,m}$ is aggregated from the influential neighbors' embeddings $\{\mathbf{e}_{t,m,k}\}$ according to their influence strengths $\{\alpha^U_{t,m,k}\}$:

$$\mathbf{c}_{t,m} = \sum\nolimits_{k=1}^{K_m} \alpha^U_{t,m,k}\mathbf{e}_{t,m,k} \tag{9}$$

S2: The ICE $\mathbf{r}_{t,m}$ w.r.t. the first-order user $u_{t,m}$ is calculated as follows:

$$\mathbf{r}_{t,m} = g_{t,m}\mathbf{c}_{t,m} + (1-g_{t,m})\mathbf{e}_{t,m} \tag{10}$$

where g measures the influence strength from the second-order neighbors. The influential gate g is modeled by a gate neural network:

$$g_{t,m} = \sigma\Big(isr_\theta\big(\mathbf{W}^{(3)}\tanh\big(\mathbf{W}^{(4)}\mathbf{c}_{t,m} + \mathbf{W}^{(5)}\mathbf{e}_{t,m}\big)\big)\Big) \tag{11}$$

where $\sigma(z) = 1/(1+e^{-z})$, $\mathbf{W}^{(4)}, \mathbf{W}^{(5)} \in \mathbb{R}^{L\times L}$ and $\mathbf{W}^{(3)} \in \mathbb{R}^{1\times L}$.

The First Level ICAU When we obtain the ICEs $\{\mathbf{r}_{t,m}\}$ w.r.t. all first-order influential neighbors, another ICAU at the first level is used to learn the target ICE:

$$\mathbf{c}^U_t = \sum\nolimits_{m=1}^{M} \alpha^U_{t,m}\mathbf{r}_{t,m} \tag{12}$$

where $\alpha^U_{t,m}$ denotes the influence strength w.r.t. the first-order ICE $\mathbf{r}_{t,m}$, which is calculated by another three-layer attention network with the same form as Eqs. 6 and 7. Then, we get the ICE $\mathbf{r}^U_t$ w.r.t. the target user u_t:

$$\mathbf{r}^U_t = g^U_t\mathbf{c}^U_t + (1-g^U_t)\mathbf{e}_t \tag{13}$$

where g^U_t is learned by a gate neural network which has the same structure with Eq. 11.

Item's Influential Context Embedding

Given an IIC $\mathcal{C}_i = \{\mathcal{I}_i, \mathcal{R}_i\}$, $\mathcal{R}_i$ is often built with the item relevance. Different from the indirect influence in user-user relation modeling, a user normally only consider those items directly relevant to the target item when they make a choice. Therefore, we only consider the first-order influential neighbors of a target item for modeling IIC. Given $\{i_{t,n}\}_{1\le n\le N}$ w.r.t. target item i_t, their corresponding embeddings $\mathbf{v}_t$ and $\{\mathbf{v}_{t,n}\}_{1\le n\le N}$ are retrieved by Item Representer E_I. As a result, we use an ICAU to learn the item ICE. The subsidiary influence embedding $\mathbf{c}^I_t$ is calculated as:

$$\mathbf{c}^I_t = \sum\nolimits_{n=1}^{N} \alpha^I_{t,n}\mathbf{v}_{t,n} \tag{14}$$

where the influential strength $\alpha^I_{t,n}$ of $\mathbf{v}_{t,n}$ is calculated by a three-layer attention network as in the UIC Aggregator.

$$\alpha^I_{t,n} = \mathrm{softmax}\Big(isr_\theta\big(\mathbf{W}^{(6)}\tanh\big(\mathbf{W}^{(7)}\mathbf{v}_{t,n}\big)\big)\Big) \tag{15}$$

where $\mathbf{W}^{(6)} \in \mathbb{R}^{L\times L}$ and $\mathbf{W}^{(7)} \in \mathbb{R}^{1\times L}$.

Subsequently, the ICE $\mathbf{r}_t^I$ w.r.t. i_t is obtained by aggregating the subsidiary influence embedding $\mathbf{c}_t^I$ and the target item embedding $\mathbf{v}_t$:

$$\mathbf{r}_t^I = g_t^I \mathbf{c}_t^I + (1 - g_t^I)\mathbf{v}_t \tag{16}$$

where the influential gate g_t^I is also learned from a three-layer gate neural network as in Figure 3.

$$g_t^I = \sigma\Big(isr_\theta\big(\mathbf{W}^{(8)} \tanh\big(\mathbf{W}^{(9)}\mathbf{c}_t^I + \mathbf{W}^{(10)}\mathbf{v}_t\big)\big)\Big) \tag{17}$$

where $\mathbf{W}^{(8)}, \mathbf{W}^{(9)} \in \mathbb{R}^{L\times L}$ and $\mathbf{W}^{(10)} \in \mathbb{R}^{1\times L}$.

User-item Interaction Ranking

Each user-item connection denotes a user's selection on an item. User-item connections can be regarded as one-class preference data (Hu et al. 2016) which cannot differentiate user preferences. To handle the one-class problem, we treat the learning on the user-item interactions as a ranking problem (Rendle et al. 2009). Given a user u_t, we construct a contrastive item pair to specify the preference order. A positive item i_p is the one which has an observed connection to u_t w.r.t. user-item relations, i.e., a user-selected item, while a pseudo-negative item i_n refers to the one without a connection to u_t. Then, we have the preference order $\langle u_t, i_p\rangle \succeq \langle u_t, i_n\rangle$. Accordingly, we have $S_{\langle u_t,i_p\rangle} \geq S_{\langle u_t,i_n\rangle}$ where $S_{\langle u_t,i\rangle}$ denotes the preference score on item i by the inner product of $\mathbf{r}_t$ and $\mathbf{r}_i$:

$$S_{\langle u_t,i_t\rangle} = \mathbf{r}_t^{U\top}\mathbf{r}_t^I \tag{18}$$

Then, we use the max-margin loss (LeCun et al. 2006) to optimize the ranking order over pairs:

$$L_{\langle u_t,i_p\rangle\succeq\langle u_t,i_n\rangle} = \max\{0, m - S_{\langle u_t,i_p\rangle} + S_{\langle u_t,i_n\rangle}\} \tag{19}$$

where $m = 10$ is set as the maximum margin in this paper.

Remark for Scoring Model: if we expand Eq. 18, we obtain the following form by using Eqs. 13 and 16:

$$\begin{aligned} S_{\langle u_t,i_t\rangle} &= \big(g_t^U\mathbf{c}_t^U + (1-g_t^U)\mathbf{e}_t\big)^\top\big(g_t^I\mathbf{c}_t^I + (1-g_t^I)\mathbf{v}_t\big) \\ &= (1-g_t^U)(1-g_t^I)\mathbf{e}_t^\top\mathbf{v}_t + g_t^U(1-g_t^I)\mathbf{c}_t^{U\top}\mathbf{v}_t \\ &\quad + (1-g_t^U)g_t^I\mathbf{e}_t^\top\mathbf{c}_t^I + g_t^U g_t^I\mathbf{c}_t^{U\top}\mathbf{c}_t^I \end{aligned} \tag{20}$$

According to Eq. 1, we find that (i) $s_{\langle u,i\rangle}$ is modeled by $\mathbf{e}_t^\top\mathbf{v}_t$; (ii) $s_{\langle u,\mathcal{I}_i^c\rangle}$ is modeled by $\mathbf{e}_t^\top\mathbf{c}_t^I$; (iii) $s_{\langle U_u^c,i\rangle}$ is modeled by $\mathbf{c}_t^{U\top}\mathbf{v}_t$, and (iv) $s_{\langle U_u^c,\mathcal{I}_i^c\rangle}$ is modeled by $\mathbf{c}_t^{U\top}\mathbf{c}_t^I$. Correspondingly, $\lambda_1 = (1-g_t^U)(1-g_t^I)$, $\lambda_2 = g_t^U(1-g_t^I)$, $\lambda_3 = (1-g_t^U)g_t^I$ and $\lambda_4 = g_t^U g_t^I$ are learned to weigh these scores. Therefore, the above-expanded terms provide an insight into how the influences in the UIC and the IIC are embedded in our model to affect the final recommendation.

Training Procedure

For each user selection $\langle u_t, i_p\rangle$, we can construct a triplet $\langle u_t, i_p, i_n\rangle$ to optimize the ranking loss $L_{\langle u_t,i_p\rangle\succeq\langle u_t,i_n\rangle}$. Then the loss of a mini-batch $\mathcal{B}$ for training is given as:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|}\sum_{\langle u_t,i_p,i_n\rangle\in\mathcal{B}} L_{\langle u_t,i_p\rangle\succeq\langle u_t,i_n\rangle} \tag{21}$$

Table 1: Statistics of the datasets: Delicious and Lastfm

	Property	User-user	Item-item	User-Item
Delicous	#Entity	1,892	17,632	1,892+17,632
	#Link	25,434	199,827	104,799
	#Link/#Entity	13.44	22.66	5.37
	Sparsity	0.0071	0.0006	0.0031
Lastfm	#Entity	1,867	69,226	1,867+69,226
	#Link	15,328	682,314	92,834
	#Link/#Entity	8.24	15.75	3.03
	Sparsity	0.0044	0.0001	0.0007

To learn the parameters, we adopt a gradient decent-based algorithm over $\partial\mathcal{L}/\partial\mathbf{W}$ w.r.t. each weight matrix $\mathbf{W}$ in our model. We implement our model using Keras (Chollet and others 2015) with Tensorflow GPU version as backend. We use Adam (Kingma and Ba 2014) as the gradient optimizer and the mini-batch size is set to 200. The code is available at: https://github.com/rainmilk/aaai19hershttps://github.com/rainmilk/aaai19hers.

Experiments

In this section, we conduct experiments on two real datasets to compare the recommendation quality of our approach with other state-of-the-art recommendation methods.

Data Preparation

Most public datasets for recommendation only involve the user-item relation, so it is not easy to find a dataset containing all user-user, user-item and item-item relations. Fortunately, two datasets, Delicious[1] and Lastfm[2], provided by RecSys Challenge 2011 (Cantador, Brusilovsky, and Kuflik 2011) can satisfy our requirement.

The Delicious dataset contains social networking, bookmarking, and tagging information from the Delicious social bookmarking system. Contact relationships are identified between users when they are mutual fans in Delicious, which is used as the user-user relation. The user-item relationships are constructed from users and their bookmarked items. The item-item relationships are built on the common tags between items. Given a target item, we assign links to top 10 items which have the most common tags.

The Lastfm dataset contains social networking, tagging, and music artist information from the Last.fm online music system. The friendships between users are used as the user-user relation. The items are artists who are connected with users if the artists' musics are listened by these users. The listening relationships between users and artists are served as the user-item relation. The item-item relationships are built through the tags using the same method as Delicious. The statistics of the datasets are summarized in Table 1 which contains the number of entities (i.e., users, items), the number of links (i.e., user-user relationships, item-item relationships, user-item interactions), the number of average links per entity, and the sparsity for each type of relations.

[1]http://www.delicious.com

[2]http://www.last.fm

Table 2: Item recommendation for test users of Delicious and Lastfm

	Delicious				Lastfm			
	MAP@5	MAP@20	nDCG@5	nDCG@20	MAP@5	MAP@20	nDCG@5	nDCG@20
BPR-MF	0.4157	0.3225	0.4318	0.3744	0.5154	0.4586	0.6252	0.6334
SoRec	0.4174	0.3390	0.4476	0.3965	0.5350	0.4775	0.6412	0.6457
Social MF	0.4181	0.3409	0.4520	0.4017	0.5489	0.4907	0.6544	0.6575
SoReg	0.4239	0.3444	0.4577	0.4056	0.5495	0.4878	0.6548	0.6541
CMF	0.4375	0.3507	0.4739	0.4158	0.5530	0.4928	0.6549	0.6749
FM	0.4246	0.3363	0.4522	0.3896	0.5366	0.4837	0.6453	0.6723
NFM	0.4565	0.3754	0.4924	0.4347	0.5462	0.4885	0.6516	0.6702
ICAU-HERS	**0.5477**	**0.4200**	**0.6064**	**0.5273**	**0.5865**	**0.5302**	**0.6913**	**0.7021**

As both datasets are very sparse, the additional information leveraged from the social networks and the item-item relations will benefit the recommendation.

Experimental Settings

To evaluate the ranking accuracy, we adopt two metrics: *Mean Average Precision* (MAP) and *normalized Discounted Cumulative Gain* (nDCG), to measure the quality of preference ranking and top-N recommendation.

Comparison Methods: The following representative methods are selected for comparison:

- *BPR-MF* (Rendle et al. 2009): it uses the BPR-based optimization on the user-item relation by MF.
- *SoRec* (Ma et al. 2008): it jointly factorizes the social-relation matrix and a user-item interaction matrix.
- *SoicalMF* (Jamali and Ester 2010): it adds regularization into MF according to social information.
- *SoReg* (Ma et al. 2011): it leverages social relationships to regularize the user's latent factors.
- *CMF* (Singh and Gordon 2008): it is adopted to co-factorize the user-user, user-item and item-item relations.
- *FM* (Rendle 2012): it embeds features into a latent space and models the interactions between each pair of features. We integrate user-user, user-item, item-item relations as the features.
- *NFM* (He and Chua 2017): a deep neural network-based FM with multiple hidden layers to learn non-linear interactions.
- *ICAU-HERS*: ICAU-based HERS proposed in this paper.

Parameter Settings: The lengths of user/item embeddings and context embeddings are set to 128. To accelerate the model, we choose top 10 first-order neighbors for each target user and 10 second-order neighbors for each first-order neighbor, and 10 neighbors for each item in the item-item relation ranked by the influence weights, cf. Eq. 7. The θ in Eq. 19 is set to 16.

Recommendation Performance

We construct the testing set by holding out 20% user-item interactions as the ground truth. For each hold-out test sample in the testing sets, we randomly draw ten noisy samples to test whether the testing methods can successfully rank the true test sample at a top position out of the noisy samples. The remaining data of the user-item relation is used for training, together with the user-user and the item-item relations.

Overall Comparison: Table 2 reports MAP and nDCG at top 5 and 10 over all testing users. Among all methods, ICAU-HERS achieves the best performance in terms of all metrics on both datasets. Specifically, ICAU-HERS demonstrates approximately 20% improvement over the second-place method NFM on Delicious and 6.1% improvement over the second-place method CMF on Lastfm in terms of MAP@5.

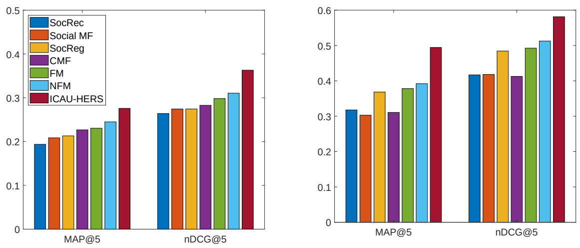
Effect of User-user Relation Modeling: BPR-MF more easily suffers from data sparsity than other comparison methods, because it cannot borrow information from the social relationships. As a result, it achieves the worst performance. In comparison, other methods benefit from the information in the user social network. Especially, ICAU-HERS outperforms all other methods, which should thank to ICAU for precisely weighing the influence from different users and aggregating high-order influences.

Effect of Item-item Relation Modeling: CMF, FM and NFM consider the item-item relation, which makes the recommendation more effective than the social relation only methods. Compared with MF and FM-based methods, ICAU-HERS demonstrates its superiority on integrating heterogeneously relational recommendation data with the influence aggregation modeling.

Recommendation for Cold-start Users and Items

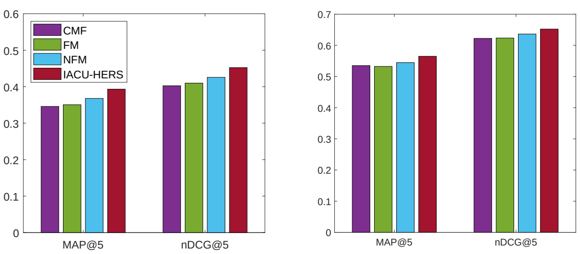
The cold-start problem is ubiquitous in RSs which can be categorized into cold-start users and cold-start items. In the user cold-start problem, for new users, we recommend items to them. The ranks over the recommended items are regarded as the evaluation criteria. In the item cold-start problem, for new items, we recommend them to different users. The ranks over the recommended users are used to evaluate the performance. In this section, we show the ICAU-HERS ability against comparison methods of handling user and item cold-start problem respectively.

Cold-start Users: To test the recommendation for cold-start users, we randomly remove 20% users and all their links from the user-item relation as the training set. BPR-MF only considers the user-item relation, so it cannot be used in the cold-start scenario. The performance compared



(a) Delicious (b) Lastfm

Figure 4: Item recommendation for cold-start users of Delicious and Lastfm



(a) Delicious (b) Lastfm

Figure 5: User recommendation for cold-start items of Delicious and Lastfm

with other methods are shown in Figure 4. ICAU-HERS significantly outperforms other methods on both datasets. On Delicious, in terms of MAP@5, the relative improvement of ICAU-HERS over SoRec, Social MF, SoReg, CMF, FM and NFM is 42.3%, 32.2%, 29.4%, 21.6%, 19.7% and 12.6% respectively. On Lastfm, in terms of MAP@5, the relative improvement of ICAU-HERS over SoRec, Social MF, SoReg, CMF, FM and NFM is 55.1%, 58.2%, 36.6%, 54.2% 30.7% and 26.1% respectively. Eq. 20 gives the insight into why ICAU-HERS can handle this cold-start problem more effectively. This is because the ICEs of users have embedded the information from their neighbors in the influential context, even when no historical user selection is observed.

Cold-start Items: To cope with cold-start items, we randomly remove 20% items and all their connected edges from the user-item relation. For each testing item, we rank the predictive scores over users. The MAP and nDCG results are shown in Figure 5. Since SoRec, Social MF and SoReg only model the user-user relation, they cannot handle cold-start items. Compared with CMF, FM and NFM, ICAU-HERS achieves the best performance on both datasets. Note that ICAU-HERS's performance on cold-start items is not as significant as that for cold-start users. This can be interpreted that the impact from other users is higher than the impact from relevant items on user's item selection. However, the embedded influential item context can still provide useful information when no selection is available for a new item.

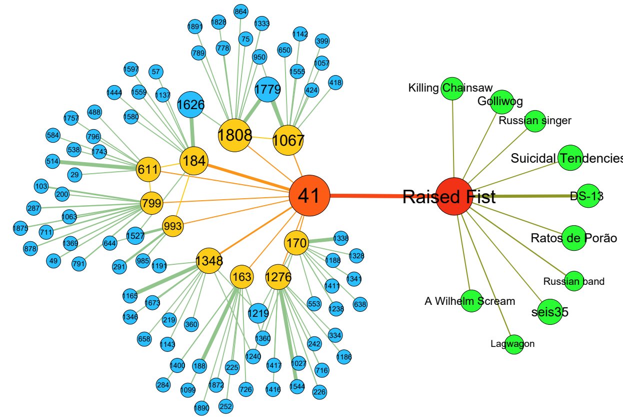


Figure 6: The visualization of influential contexts of a sampled user selection on an artist in the Lastfm dataset. The artists in the item network are labeled by their names and the anonymous users in the user network are labeled with their IDs. The thickness of edges specifies the significance of influence.

Visualization and Interpretation

To interpret the influence from influential users and items when a user makes an item selection, we randomly choose a user and her/his selection on an artist (User 41 with Raised Fist) from the LastFM dataset as a case study. We visualize the influential context w.r.t. the target user and the target item in Figure 6 by differentiating the influence with different edge thickness. We can find that the influence from different users/items is quite different, for example, User 1808 and User 184 have the common neighbor 1626 but the influence of User 1626 on User 184 is much larger than that on User 1808. In the item network, we observe similar cases from the influential context w.r.t. Raised Fist. Therefore, these influential contexts can provide the interpretation of how the target users and the target items are influenced by the influential users and items to form the connection.

Conclusion

We propose a framework for modeling influential contexts in recommender systems with the user-user, the item-item and the user-item relations, and further implement a neural ICAU-based HERS model. In particular, this HERS can effectively learn the ICEs from user's and item's influential contexts through ICAU. Extensive experiments show that ICAU-HERS outperforms other state-of-the-art methods and it is effective for handling the user/item cold-start and sparsity problems. In addition, we demonstrate the interpretability of ICAU-HERS with a real case study.

ICAU is a general influence embedding model which can be applied to other domains with heterogeneous networks, such as user group behavior analysis and biological interaction network, apart from recommender systems. Moreover, it is easy to incorporate content information of each entity to better interpret the influence propagation (Jian et al. 2019).

Acknowledgements

The paper is partially supported by the National Natural Science Foundation of China (61572325); Shanghai Key Programs of Science and Technology(16DZ1203603); Shanghai Engineering Research Center Project (GCZX14014 and C14001).

References

Cantador, I.; Brusilovsky, P.; and Kuflik, T. 2011. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *HetRec 2011*.

Cao, L. 2015. Coupling learning of complex interactions. *Information Processing & Management* 51(2):167–186.

Cao, L. 2016. Non-iid recommender systems: A review and framework of recommendation paradigm shifting. *Engineering* 2(2):212–224.

Chollet, F., et al. 2015. Keras. https://github.com/fchollet/keras.

Do, T. D. T., and Cao, L. 2018a. Coupled poisson factorization integrated with user/item metadata for modeling popular and sparse ratings in scalable recommendation. In *AAAI'2018*, 2918–2925.

Do, T. D. T., and Cao, L. 2018b. Gamma-poisson dynamic matrix factorization embedded with metadata influence. In *NIPS'2018*.

Friedkin, N. E., and Johnsen, E. C. 2011. *Social influence network theory: A sociological examination of small group dynamics*, volume 33. Cambridge University Press.

He, X., and Chua, T.-S. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR'2017*, 355–364.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *WWW'2017*, 173–182.

Hu, L.; Cao, J.; Xu, G.; Cao, L.; Gu, Z.; and Cao, W. 2014. Deep modeling of group preferences for group-based recommendation. In *AAAI'2014*, 1861–1867.

Hu, L.; Cao, L.; Cao, J.; Gu, Z.; Xu, G.; and Yang, D. 2016. Learning informative priors from heterogeneous domains to improve recommendation in cold-start user domains. *ACM Transactions on Information Systems (TOIS)* 35(2):13.

Hu, L.; Cao, L.; Cao, J.; Gu, Z.; Xu, G.; and Wang, J. 2017a. Improving the quality of recommendations for users and items in the tail of distribution. *ACM Transactions on Information Systems (TOIS)* 35(3):25.

Hu, L.; Cao, L.; Wang, S.; Xu, G.; Cao, J.; and Gu, Z. 2017b. Diversifying personalized recommendation with user-session context. In *IJCAI'2017*, 1858–1864.

Jamali, M., and Ester, M. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys'2010*, 135–142. ACM.

Jian, S.; Hu, L.; Cao, L.; Lu, K.; and Gao, H. 2019. Evolutionarily learning multi-aspect interactions and influences from network structure and node content. In *AAAI'2019*.

Jiang, M.; Cui, P.; Wang, F.; Zhu, W.; and Yang, S. 2014. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering* 26(11):2789–2802.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; and Huang, F. 2006. A tutorial on energy-based learning. *Predicting Structured Data* 1:0.

Li, F.; Xu, G.; and Cao, L. 2015. Coupled matrix factorization within non-iid context. In *PAKDD'2015*, 707–719.

Ma, H.; Yang, H.; Lyu, M. R.; and King, I. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM'2008*, 931–940. ACM.

Ma, H.; Zhou, D.; Liu, C.; Lyu, M. R.; and King, I. 2011. Recommender systems with social regularization. In *WSDM'2011*, 287–296. ACM.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI'2009*, 452–461.

Rendle, S. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3(3):57.

Rendle, S. 2013. Scaling factorization machines to relational data. In *Proceedings of the VLDB Endowment*, volume 6, 337–348. VLDB Endowment.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *KDD'2008*, 650–658.

Tang, J.; Gao, H.; Liu, H.; and Das Sarma, A. 2012. etrust: Understanding trust evolution in an online world. In *KDD'2012*, 253–261.

Tang, J.; Hu, X.; and Liu, H. 2013. Social recommendation: a review. *Social Network Analysis and Mining* 3(4):1113–1133.

Vasuki, V.; Natarajan, N.; Lu, Z.; and Dhillon, I. S. 2010. Affiliation recommendation using auxiliary networks. In *RecSys'2010*, 103–110.

Wang, X.; He, X.; Nie, L.; and Chua, T.-S. 2017. Item silk road: Recommending items from information domains to social users. In *SIGIR'2017*, 185–194. ACM.

Zhang, Q.; Cao, L.; Zhu, C.; Li, Z.; and Sun, J. 2018. Coupledcf: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering. In *IJCAI'2018*, 3662–3668.