Heterogeneous Information Network Embedding for Recommendation

Chuan Shi[ID], *Member, IEEE*, Binbin Hu[ID],
Wayne Xin Zhao[ID], *Member, IEEE*, and Philip S. Yu[ID], *Fellow, IEEE*

Abstract—Due to the flexibility in modelling data heterogeneity, heterogeneous information network (HIN) has been adopted to characterize complex and heterogeneous auxiliary data in recommender systems, called *HIN based recommendation*. It is challenging to develop effective methods for HIN based recommendation in both extraction and exploitation of the information from HINs. Most of HIN based recommendation methods rely on path based similarity, which cannot fully mine latent structure features of users and items. In this paper, we propose a novel heterogeneous network embedding based approach for HIN based recommendation, called HERec. To embed HINs, we design a meta-path based random walk strategy to generate meaningful node sequences for network embedding. The learned node embeddings are first transformed by a set of fusion functions, and subsequently integrated into an extended matrix factorization (MF) model. The extended MF model together with fusion functions are jointly optimized for the rating prediction task. Extensive experiments on three real-world datasets demonstrate the effectiveness of the HERec model. Moreover, we show the capability of the HERec model for the cold-start problem, and reveal that the transformed embedding information from HINs can improve the recommendation performance.

Index Terms—Heterogeneous information network, network embedding, matrix factorization, recommender system

✦

1 INTRODUCTION

IN recent years, recommender systems, which help users discover items of interest from a large resource collection, have been playing an increasingly important role in various online services [1], [2]. Traditional recommendation methods (e.g., matrix factorization) mainly aim to learn an effective prediction function for characterizing user-item interaction records (e.g., user-item rating matrix). With the rapid development of web services, various kinds of auxiliary data (*a.k.a.*, side information) become available in recommender systems. Although auxiliary data is likely to contain useful information for recommendation [3], it is difficult to model and utilize these heterogeneous and complex information in recommender systems. Furthermore, it is more challenging to develop a relatively general approach to model these varying data in different systems or platforms.

As a promising direction, heterogeneous information network (HIN), consisting of multiple types of nodes and links, has been proposed as a powerful information modeling method [4], [5], [6]. Due to its flexibility in modeling data heterogeneity, HIN has been adopted in recommender systems to characterize rich auxiliary data. In Fig. 1a, we present an example for movie recommendation characterized by HINs. We can see that the HIN contains multiple types of entities connected by different types of relations. Under the HIN based representation, the recommendation problem can be considered as a similarity search task over the HIN [4]. Such a recommendation setting is called as *HIN based recommendation* [7]. HIN based recommendation has received much attention in the literature [7], [8], [9], [10], [11], [12]. The basic idea of most existing HIN based recommendation methods is to leverage path based semantic relatedness between users and items over HINs, e.g., meta-path based similarities, for recommendation.

Although HIN based methods have achieved performance improvement to some extent, there are two major problems for these methods using meta-path based similarities. First, meta-path based similarities rely on explicit path reachability, and may not be reliable to be used for recommendation when path connections are sparse or noisy. It is likely that some links in HINs are accidentally formed which do not convey meaningful semantics. Second, meta-path based similarities mainly characterize semantic relations defined over HINs, and may not be directly applicable to recommender systems. It is likely that the derived path based similarities have no explicit impact on the recommendation performance in some cases. Existing methods mainly learn a linear weighting mechanism to combine the path based similarities [11] or latent factors [7], which cannot learn the complicated mapping mechanism of HIN information for recommendation. The two problems essentially reflect two fundamental issues for HIN based recommendation, namely effective information extraction and exploitation based on HINs for recommendation.

For the first issue, it is challenging to develop a way to effectively extract and represent useful information for

- C. Shi and B. Hu are with the Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: {shichuan, hubinbin}@bupt.edu.cn.
- W.X. Zhao is with the School of Information, Renmin University of China, Haidian 100872, China. E-mail: batmanfly@gmail.com.
- P.S. Yu is with the University of Illinois at Chicago, Chicago, IL 60607, and the Institute for Data Science, Tsinghua University, Beijing 100083, China. E-mail: psyu@cs.uic.edu.

(a) An Example of HIN   **(b) HIN Embedding**   **(c) Recomendation**
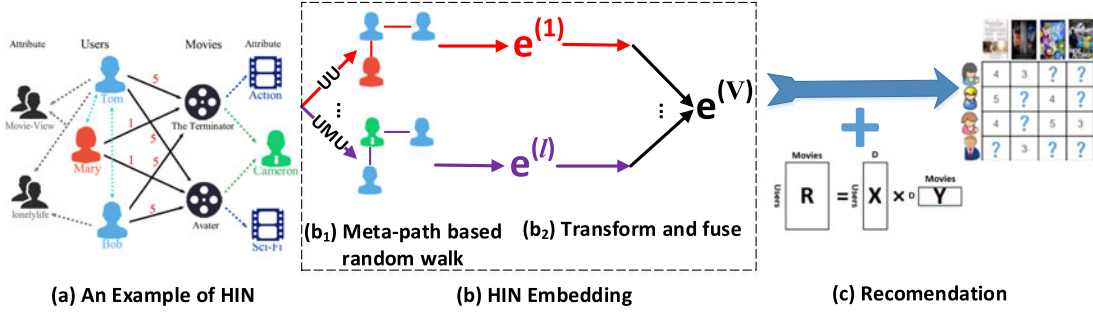
Fig. 1. The schematic illustration of the proposed HERec approach.

HINs due to data heterogeneity. Unlike previous studies using meta-path based similarities [4], [7], our idea is to learn effective heterogeneous network representations for summarizing important structural characteristics and properties of HINs. Following [13], [14], we characterize nodes from HINs with low-dimensional vectors, i.e., embeddings. Instead of relying on explicit path connection, we would like to encode useful information from HINs with latent vectors. Compared with meta-path based similarity, the learned embeddings are in a more compact form that is easy to use and integrate. Also, the network embedding approach itself is more resistant to sparse and noisy data. However, most existing network embedding methods focus on homogeneous networks only consisting of a single type of nodes and links, and cannot directly deal with heterogeneous networks consisting of multiple types of nodes and links. Hence, we propose a new heterogeneous network embedding method. Considering heterogeneous characteristics and rich semantics reflected by meta-paths, the proposed method first uses a random walk strategy guided by meta-paths to generate node sequences. For each meta-path, we learn a unique embedding representation for a node by maximizing its co-occurrence probability with neighboring nodes in the sequences sampled according to the given meta-path. We fuse the multiple embeddings *w.r.t.* different meta-paths as the output of HIN embedding.

After obtaining the embeddings from HINs, we study how to integrate and utilize such information in recommender systems. We don't assume the learned embeddings are naturally applicable in recommender systems. Instead, we propose and explore three fusion functions to integrate multiple embeddings of a node into a single representation for recommendation, including simple linear fusion, personalized linear fusion and non-linear fusion. These fusion functions provide flexible ways to transform HIN embeddings into useful information for recommendation. Specially, we emphasize that personalization and non-linearity are two key points to consider for information transformation in our setting. Finally, we extend the classic matrix factorization framework by incorporating the fused HIN embeddings. The prediction model and the fusion function are jointly optimized for the rating prediction task.

By integrating the above two parts together, this work presents a novel HIN embedding based recommendation approach, called *HERec* for short. HERec first extracts useful HIN based information using the proposed HIN embedding method, and then utilizes the extracted information for recommendation using the extended matrix factorization model. We present the overall illustration for the proposed approach in Fig. 1. Extensive experiments on three real-world datasets demonstrate the effectiveness of the proposed approach. We

also verify the ability of HERec to alleviate cold-start problem and examine the impact of meta-paths on performance. The key contributions of this paper can be summarized as follows:

- We propose a heterogeneous network embedding method guided by meta-paths to uncover the semantic and structural information of heterogeneous information networks. Moreover, we propose a general embedding fusion approach to integerate different embeddings based on different meta-paths into a single representation.
- We propose a novel heterogeneous information network embedding for recommendation model, called HERec for short. HERec can effectively integrate various kinds of embedding information in HIN to enhance the recommendation performance. In addition, we design a set of three flexible fusion functions to effectively transform HIN embeddings into useful information for recommendation.
- Extensive experiments on three real-world datasets demonstrate the effectiveness of the proposed model. Moreover, we show the capability of the proposed model for the cold-start prediction problem, and reveal that the transformed embedding information from HINs can improve the recommendation performance.

The remainder of this paper is organized as follows. Section 2 introduces the related works. Section 3 describes notations used in the paper and presents some preliminary knowledge. Then, We propose the heterogeneous network embedding method and the HERec model in Section 4. Experiments and detailed analysis are reported in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

In this section, we will review the related studies in three aspects, namely recommender systems, heterogeneous information networks and network embedding.

In the literature of recommender systems, early works mainly adopt collaborative filtering (CF) methods to utilize historical interactions for recommendation [3]. Particularly, the matrix factorization approach [15], [16] has shown its effectiveness and efficiency in many applications, which factorizes user-item rating matrix into two low rank user-specific and item-specific matrices, and then utilizes the factorized matrices to make further predictions [2]. Since CF methods usually suffer from cold-start problem, many works [8], [17], [18] attempt to leverage additional information to improve recommendation performance. For example, Ma et al. [19] integrate social relations into matrix factorization in recommendation. Ling et al. [20] consider the information of both ratings and reviews and propose a

unified model to combine content based filtering with collaborative filtering for rating prediction task. Ye et al. [21] incorporate user preference, social influence and geographical influence in the recommendation and propose a unified POI recommendation framework. More recently, Sedhain et al. [22] explain drawbacks of three popular cold-start models [23], [24], [25] and further propose a learning based approach for the cold-start problem to leverage social data via randomised SVD. And many works begin to utilize deep models (e.g., convolutional neural network, auto encoder) to exploit text information [26], image information [27] and network structure information [28] for better recommendation. In addition, there are also some typical frameworks focusing on incorporating auxiliary information for recommendation. Chen et al. [29] propose a typical SVDFeature framework to efficiently solve the feature based matrix factorization. And Rendle [30] proposes factorization machine, which is a generic approach to combine the generality of feature engineering.

As a newly emerging direction, heterogeneous information network [5] can naturally model complex objects and their rich relations in recommender systems, in which objects are of different types and links among objects represent different relations [31], [32]. And several path based similarity measures [4], [33], [34] are proposed to evaluate the similarity of objects in heterogeneous information network. Therefore, some researchers have began to be aware of the importance of HIN based recommendation. Wang et al. [8] propose the OptRank method to alleviate the cold-start problem by utilizing heterogeneous information contained in social tagging system. Furthermore, the concept of meta-path is introduced into hybrid recommender systems [35]. Yu et al. [7] utilize meta-path based similarities as regularization terms in the matrix factorization framework. Yu et al. [9] take advantage of different types of entity relationships in heterogeneous information network and propose a personalized recommendation framework for implicit feedback dataset. Luo et al. [36] propose a collaborative filtering based social recommendation method using heterogeneous relations. More recently, Shi et al. [10] propose the concept of weighted heterogeneous information network and design a meta-path based collaborative filtering model to flexibly integrate heterogeneous information for personalized recommendation. In [11], [12], [37], the similarities of users and items are both evaluated by path based similarity measures under different semantic meta-paths and a matrix factorization based on dual regularization framework is proposed for rating prediction. Besides meta-path, Zhao et al. [38] propose a factorization machine based model integrated with meta-graph based similarity for recommendation. Most of HIN based methods rely on the path based similarity, which may not fully mine latent features of users and items on HINs for recommendation.

On the other hand, network embedding has shown its potential in structure feature extraction and has been successfully applied in many data mining tasks [39], [40], [41], [42], such as classification [43], clustering [44], [45] and recommendation [46], [47]. Deepwalk [13] combines random walk and skip-gram to learn network representations. Furthermore, Grover and Leskovec [14] propose a more flexible network embedding framework based on a biased random walk procedure. In addition, LINE [48] and SDNE [49] characterize the second-order link proximity, as well as neighbor relations. Cao et al. [50] propose the GraRep model to capture higher-order graph proximity for network representations. Besides leaning network embedding from only the topology, there are also many works [51], [52], [53] leveraging node content information and other available graph information for the robust representations. Unfortunately, most of network embedding methods focus on homogeneous networks, and thus they cannot directly be applied for heterogeneous networks. Recently, several works [54], [55], [56], [57], [58] attempt to analyze heterogeneous networks via embedding methods. Particularly, Chang et al. [54] design a deep embedding model to capture the complex interaction between the heterogeneous data in the network. Xu et al. [56] propose a EOE method to encode the intra-network and inter-network edges for the coupled heterogeneous network. Dong et al. [58] define the neighbor of nodes via meta-path and learn the heterogeneous embedding by skip-gram with negative sampling. More recently, Fu et al. [59] utilize a neural network model to capture rich relation semantics in HIN. Although these methods can learn network embeddings in various heterogeneous network, their representations of nodes and relations may not be optimum for recommendation.

To our knowledge, there are few attempts which adopt the network embedding approach to extract useful information from heterogeneous information network and leverage such information for rating prediction. The proposed approach utilizes the flexibility of HIN for modeling complex heterogeneous context information, and meanwhile borrows the capability of network embedding for learning effective information representation. The final rating prediction component further incorporates a transformation mechanism implemented by three flexible functions to utilize the learned information from network embedding.

3 PRELIMINARY

A heterogeneous information network is a special kind of information network, which either contains multiple types of objects or multiple types of links.

Definition 1 (Heterogeneous information network [60]). *A HIN is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consisting of a object set $\mathcal{V}$ and a link set $\mathcal{E}$. A HIN is also associated with an object type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and a link type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the sets of predefined object and link types, where $|\mathcal{A}| + |\mathcal{R}| > 2$.*

The complexity of heterogeneous information network drives us to provide the meta level (e.g., schema-level) description for understanding the object types and link types better in the network. Hence, the concept of network schema is proposed to describe the meta structure of a network.

Definition 2 (Network schema [31], [61]). *The network schema is denoted as $\mathcal{S} = (\mathcal{A}, \mathcal{R})$. It is a meta template for an information network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the object type mapping $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and the link type mapping $\psi : \mathcal{E} \rightarrow \mathcal{R}$, which is a directed graph defined over object types $\mathcal{A}$, with edges as relations from $\mathcal{R}$.*

Example 1. As shown in Fig. 1a, we have represented the setting of movie recommender systems by HINs. We further present its corresponding network schema in Fig. 2a, consisting of multiple types of objects, including User (U), Movie (M), Director (D). There exist different types of links between objects to represent different relations. A user-user link indicates the friendship between two users, while a

(a) Douban Movie   (b) Douban Book   (c) Yelp
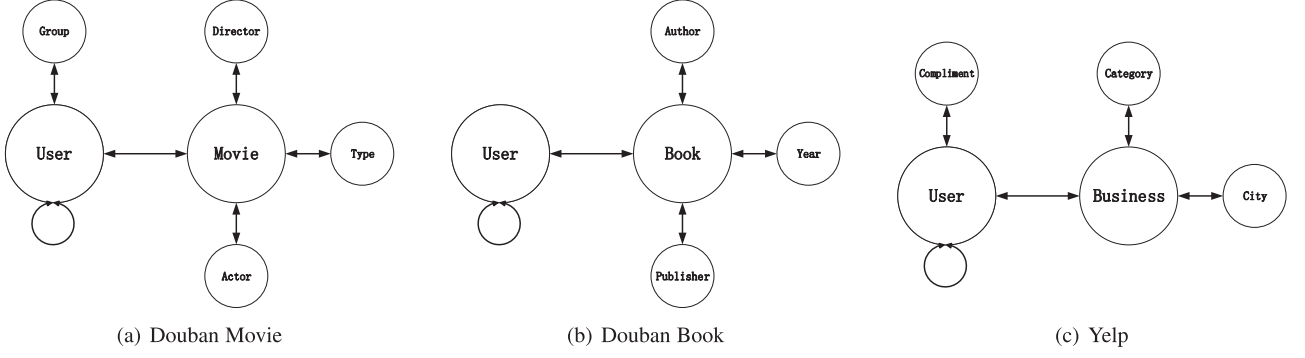
Fig. 2. Network schemas of heterogeneous information networks for the used three datasets. In our task, users and items are our major focus, denoted by large-sized circles, while the other attributes are denoted by small-sized circles.

user-movie link indicates the rating relation. Similarly, we present the schematic network schemas for book and business recommender systems in Fig. 2b and 2c respectively.

In HINs, two objects can be connected via different semantic paths, which are called meta-paths.

Definition 3 (Meta-path [4]). *A meta-path ρ is defined on a network schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ and is denoted as a path in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ (abbreviated as $A_1 A_2 \cdots A_{l+1}$), which describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between object A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations.*

Example 2. Taking Fig. 2a as an example, two objects can be connected via multiple meta-paths, e.g., "User—User" (*UU*) and "User—Movie—User" (*UMU*). Different meta-paths usually convey different semantics. For example, the *UU* path indicates friendship between two users, while the *UMU* path indicates the co-watch relation

TABLE 1
Notations and Explanations

Notation	Explanation
$\mathcal{G}$	heterogeneous information network
$\mathcal{V}$	object set
$\mathcal{E}$	link set
$\mathcal{S}$	network schema
$\mathcal{A}$	object type set
$\mathcal{R}$	link type set
$\mathcal{U}$	user set
$\mathcal{I}$	item set
$\widehat{r_{u,i}}$	predicted rating user u gives to item i
e_v	low-dimensional representation of node v
$\mathcal{N}_u$	neighborhood of node u
ρ	a meta-path
$\mathcal{P}$	meta-path set
$e_u^{(U)}, e_i^{(I)}$	final representations of user u, item i
d	dimension of HIN embeddings
D	dimension of latent factors
$\mathbf{x}_u, \mathbf{y}_i$	latent factors of user u, item i
$\gamma_u^{(U)}, \gamma_i^{(I)}$	latent factors for pairing HIN embedding of user u, item i
α, β	parameters for integrating HIN embeddings
$\mathbf{M}^{(l)}$	transformation matrix w.r.t the lth mete-path
$\boldsymbol{b}^{(l)}$	bias vector w.r.t the lth mete-path
$w_u^{(l)}$	preference weight of user u over the lth meta-path
$\boldsymbol{\Theta}^{(U)}, \boldsymbol{\Theta}^{(I)}$	parameters of fusion functions for users, items
λ	regularization parameter
η	learning rate

between two users, i.e., they have watched the same movies. As will be seen later, the detailed meta-paths used in this work is summarized in Table 2.

Recently, HIN has become a mainstream approach to model various complex interaction systems [5]. Specially, it has been adopted in recommender systems for characterizing complex and heterogenous recommendation settings.

Definition 4 (HIN based recommendation). *In a recommender system, various kinds of information can be modeled by a HIN $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. On recommendation-oriented HINs, two kinds of entities (i.e., users and items) together with the relations between them (i.e., rating relation) are our focus. Let $\mathcal{U} \subset \mathcal{V}$ and $\mathcal{I} \subset \mathcal{V}$ denote the sets of users and items respectively, a triplet $\langle u, i, r_{u,i} \rangle$ denotes a record that a user u assigns a rating of $r_{u,i}$ to an item i, and $\mathcal{R} = \{\langle u, i, r_{u,i} \rangle\}$ denotes the set of rating records. We have $\mathcal{U} \subset \mathcal{V}, \mathcal{I} \subset \mathcal{V}$ and $\mathcal{R} \subset \mathcal{E}$. Given the HIN $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the goal is to predict the rating score $r_{u,i'}$ of $u \in \mathcal{U}$ to a non-rated item $i' \in \mathcal{I}$.*

Several efforts have been made for HIN based recommendation. Most of these works mainly leverage meta-path based similarities to enhance the recommendation performance [7], [9], [10], [11]. Next, we will present a new heterogeneous network embedding based approach to this task, which is able to effectively exploit the information reflected in HINs. The notations we will use throughout the article are summarized in Table 1.

4 THE PROPOSED APPROACH

In this section, we present a Heterogeneous network Embedding based approach for Recommendation, called *HERec*. For addressing the two issues introduced in Section 1, the proposed HERec approach consists of two major components. First, we propose a new heterogeneous network embedding method to learn the user/item embeddings from HINs. Then, we extend the classic matrix factorization framework by incorporating the learned embeddings using a flexible set of fusion functions. We present an overall schematic illustration of the proposed approach in Fig. 1. After the construction of the HINs (Fig. 1a), two major steps are presented, namely HIN embedding (Fig. 1b) and recommendation (Fig. 1c). Next, we will present detailed illustration of the proposed approach.

4.1 Heterogeneous Network Embedding

Inspired by the recent progress on network embedding [13], [14], we adopt the representation learning method to extract

TABLE 2
Statistics of the Three Datasets

Dataset (Density)	Relations (A-B)	Number of A	Number of B	Number of (A-B)	Ave. degrees of A	Ave. degrees of B	Meta-paths
Douban Movie (0.63%)	User-Movie	13,367	12,677	1,068,278	79.9	84.3	UMU, MUM
	User-User	2,440	2,294	4,085	1.7	1.8	UMDMU, MDM
	User-Group	13,337	2,753	570,047	42.7	207.1	UMAMU, MAM
	Movie-Director	10,179	2,449	11,276	1.1	4.6	UMTMU, MTM
	Movie-Actor	11,718	6,311	33,587	2.9	5.3	
	Movie-Type	12,678	38	27,668	2.2	728.1	
Douban Book (0.27%)	User-Book	13,024	22,347	792,026	60.8	35.4	UBU, BUB
	User-User	12,748	12,748	169,150	13.3	13.3	UBPBU, BPB
	Book-Author	21,907	10,805	21,905	1.0	2.0	UBYBU, BYB
	Book-Publisher	21,773	1,815	21,773	1.0	11.9	UBABU
	Book-Year	21,192	64	21,192	1.0	331.1	
Yelp (0.08%)	User-Business	16,239	14,284	198,397	12.2	13.9	UBU, BUB
	User-User	10,580	10,580	158,590	15.0	15.0	UBCiBU, BCiB
	User-Compliment	14,411	11	76,875	5.3	6988.6	UBCaBU, BCaB
	Business-City	14,267	47	14,267	1.0	303.6	
	Business-Category	14,180	511	40,009	2.8	78.3	

and represent useful information of HINs for recommendation. Given a HIN $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, our goal is to learn a low-dimensional representation $e_v \in \mathbb{R}^d$ (a.k.a., embedding) for each node $v \in \mathcal{V}$. The learned embeddings are expected to highly summarize informative characteristics, which are likely to be useful in recommender systems represented on HINs. Compared with meta-path based similarity [4], [10], it is much easier to use and integrate the learned representations in subsequent procedures.

However, most of the existing network embedding methods mainly focus on homogeneous networks, which are not able to effectively model heterogeneous networks. For instance, the pioneering study *deepwalk* [13] uses random walk to generate node sequences, which cannot discriminate nodes and edges with different types. Hence, it requires a more principled way to traverse the HINs and generate meaningful node sequences.

4.1.1 Meta-Path Based Random Walk

To generate meaningful node sequences, the key is to design an effective walking strategy that is able to capture the complex semantics reflected in HINs. In the literature of HINs, meta-path is an important concept to characterize the semantic patterns for HINs [4]. Hence, we propose to use the meta-path based random walk method to generate node sequences. Giving a heterogeneous network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and a meta-path $\rho : A_1 \xrightarrow{R_1} \cdots A_t \xrightarrow{R_t} A_{t+1} \cdots \xrightarrow{R_l} A_{l+1}$, the walk path is generated according to the following distribution:

$$P(n_{t+1} = x | n_t = v, \rho)$$
$$= \begin{cases} \frac{1}{|\mathcal{N}^{A_{t+1}}(v)|}, & (v, x) \in \mathcal{E} \text{ and } \phi(x) = A_{t+1}; \\ 0, & \text{otherwise}, \end{cases} \quad (1)$$

where n_t is the tth node in the walk, v has the type of A_t, and $\mathcal{N}^{(A_{t+1})}(v)$ is the first-order neighbor set for node v with the type of A_{t+1}. A walk will follow the pattern of a meta-path repetitively until it reaches the pre-defined length.

Example 3. We still take Fig. 1a as an example, which represents the heterogeneous information network of movie

recommender systems. Given a meta-path UMU, we can generate two sample walks (i.e., node sequences) by starting with the user node of *Tom*: (1) $\text{Tom}_{User} \to \text{The Terminator}_{Movie} \to \text{Mary}_{User}$, and (2) $\text{Tom}_{User} \to \text{Avater}_{Movie} \to \text{Bob}_{User} \to \text{The Terminator}_{Movie} \to \text{Mary}_{User}$. Similarly, given the meta-path $UMDMU$, we can also generate another node sequence: $\text{Tom}_{User} \to \text{The Terminator}_{Movie} \to \text{Cameron}_{Director} \to \text{Avater}_{Movie} \to \text{Mary}_{User}$. It is intuitive to see that these meta-paths can lead to meaningful node sequences corresponding to different semantic relations.

4.1.2 Type Constraint and Filtering

Since our goal is to improve the recommendation performance, the main focus is to learn effective representations for users and items, while objects with other types are of less interest in our task. Hence, we only select meta-paths starting with *user type* or *item type*. Once a node sequence has been generated using the above method, it is likely to contain nodes with different types. We further remove the nodes with a type different from the starting type. In this way, the final sequence will only consist of nodes with the starting type. Applying type filtering to node sequences has two benefits. First, although node sequences are constructed using meta-paths with heterogeneous types, the final representations are learned using homogeneous neighborhood. We embed the nodes with the same type in the same space, which relaxes the challenging goal of representing all the heterogeneous objects in a unified space. Second, given a fixed-length window, a node is able to utilize more homogeneous neighbors that are more likely to be relevant than others with different types.

Example 4. As shown in Fig. 3, in order to learn effective representations for users and items, we only consider the meta-paths in which the starting type is *user type* or *item type*. In this way, we can derive some meta-paths, such as UMU, $UMDMU$ and MUM. Take the meta-path of UMU as an instance. We can generate a sampled sequence "$u_1 \to m_1 \to u_2 \to m_2 \to u_3 \to m_2 \to u_4$" according to Eq. (1). Once a sequence has been constructed, we further remove the nodes with a different type compared with
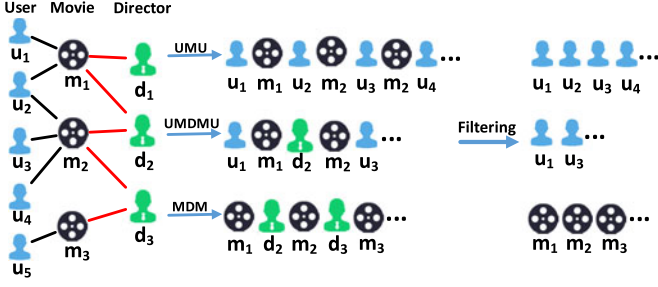
Fig. 3. An illustrative example of the proposed meta-path based random walk. We first perform random walks guided by some selected meta-paths, and then filter the node sequences *not with* the user type or item type.

the starting node. In this way, we finally obtain a homogeneous node sequence "$u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4$".

The connections between homogeneous nodes are essentially constructed via the heterogeneous neighborhood nodes. After this step, our next focus will be how to learn effective representations for homogeneous sequences.

4.1.3   Optimization Objective

Given a meta-path, we can construct the neighborhood $\mathcal{N}_u$ for node u based on co-occurrence in a fixed-length window. Following node2vec [14], we can learn the representations of nodes to optimize the following objective:

$$\max_f \sum_{u \in \mathcal{V}} \log Pr(\mathcal{N}_u | f(u)), \quad (2)$$

where $f : \mathcal{V} \rightarrow \mathbb{R}^d$ is a function (aiming to learn) mapping each node to d-dimensional feature space, and $\mathcal{N}_u \subset \mathcal{V}$ represents the neighborhood of node u, *w.r.t.* a specific meta-path. We can learn the embedding mapping function $f(\cdot)$ by applying stochastic gradient descent (SGD) to optimize this objective. A major difference between previous methods and ours lies in the construction of $\mathcal{N}_u$. Our method selects homogeneous neighbors using meta-path based random walks.

4.1.4   Embedding Fusion

Heterogeneous network embedding provides a general way to extract useful information from HINs. For our model, given a node $v \in \mathcal{V}$, we can obtain a set of representations $\{e_v^{(l)}\}_{l=1}^{|\mathcal{P}|}$, where $\mathcal{P}$ denotes the set of meta-paths, and $e_v^{(l)}$ denotes the representation of v *w.r.t.* the lth meta-path. It requires a principled fusion way to transform node embeddings into a more suitable form that is useful to improve recommendation performance. Existing studies usually adopt a linear weighting mechanism to combine the information mined from HINs (e.g., meta-path based similarities), which may not be capable of deriving effective information representations for recommendation. Hence, we propose to use a general function $g(\cdot)$, which aims to fuse the learned node embeddings for users and items:

$$
\begin{aligned}
e_u^{(U)} &\leftarrow g(\{e_u^{(l)}\}), \\
e_i^{(I)} &\leftarrow g(\{e_i^{(l)}\}),
\end{aligned}
\quad (3)
$$

where $e_u^{(U)}$ and $e_i^{(I)}$ are the final representations for a user u and an item i respectively, called *HIN embeddings*. Since users and items are our focus, we only learn the embeddings for users and items. At the current stage, we do not

specify the form of function $g(\cdot)$. Instead, we believe that a good fusion function should be learned according to the specific task. Hence, we leave the formulation and optimization of the fusion function in our recommendation model.

4.2   Integrating Matrix Factorization with Fused HIN Embedding for Recommendation

Previously, we have studied how to extract and represent useful information from HINs for recommendation. With HIN embedding, we can obtain user embeddings $\{e_u^{(U)}\}_{u \in \mathcal{U}}$ and item embeddings $\{e_i^{(I)}\}_{i \in \mathcal{I}}$, which are further specified by a function $g(\cdot)$ that is to learn. Now we study how to utilize the learned embeddings for recommendation.

4.2.1   Rating Predictor

We build our rating predictor based on the classic matrix factorization (MF) model [62], which factorizes the user-item rating matrix into user-specific and item-specific matrices. In MF, the rating of a user u on an item i is simply defined as follows:

$$\widehat{r_{u,i}} = \mathbf{x}_u^\top \cdot \mathbf{y}_i, \quad (4)$$

where $\mathbf{x}_u \in \mathbb{R}^D$ and $\mathbf{y}_i \in \mathbb{R}^D$ denote the latent factors corresponding to user u and item i. Since we have also obtained the representations for user u and item i, we further incorporate them into the rating predictor as below

$$\widehat{r_{u,i}} = \mathbf{x}_u^\top \cdot \mathbf{y}_i + \alpha \cdot e_u^{(U)^\top} \cdot \gamma_i^{(I)} + \beta \cdot \gamma_u^{(U)^\top} \cdot e_i^{(I)}, \quad (5)$$

where $e_u^{(U)}$ and $e_i^{(I)}$ are the fused embeddings, $\gamma_u^{(U)}$ and $\gamma_i^{(I)}$ are user-specific and item-specific latent factors to pair with the HIN embeddings $e_u^{(U)}$ and $e_i^{(I)}$ respectively, and α and β are the tuning parameters to integrate the three terms. For Eq. (5), we need to note two points. First, $e_u^{(U)}$ and $e_i^{(I)}$ are the output of function $g(\cdot)$ in Eq. (3). We assume that the derived embeddings after transformation by function $g(\cdot)$ are applicable in MF. Second, we don't directly pair $e_u^{(U)}$ with $e_i^{(I)}$. Recall the proposed embedding method indeed characterizes the relatedness between objects with the same type. We incorporate new latent factors $\gamma_u^{(U)}$ and $\gamma_i^{(I)}$ to relax the assumption that $e_u^{(U)}$ and $e_i^{(I)}$ have to be in the same space, which increases the flexibility to the prediction model.

4.2.2   Setting the Fusion Function

Previously, we assume the function $g(\cdot)$ has been given in a general form. Now, we study how to set the fusion function, which transforms HIN embeddings into a form that is useful in recommender systems. We only discuss the function for fusing user embeddings, and it is similar to fuse item embeddings. We propose to use three fusion functions to integrate embeddings:

- Simple linear fusion. We assume each user has the same preference for each meta-path, and therefore, we assign each meta-path with a unified weight (i.e., average value) for each user. Moreover, we linearly transform embeddings to target space.

$$g(\{e_u^{(l)}\}) = \frac{1}{|\mathcal{P}|} \sum_{l=1}^{|\mathcal{P}|} (\mathbf{M}^{(l)} e_u^{(l)} + \boldsymbol{b}^{(l)}), \quad (6)$$

where $\mathcal{P}$ is the set of considered meta-paths, $\mathbf{M}^{(l)} \in \mathbb{R}^{D \times d}$ and $\boldsymbol{b}^{(l)} \in \mathbb{R}^D$ are the transformation matrix and bias vector *w.r.t.* the lth meta-path.

- Personalized linear fusion. The simple linear fusion cannot model users' personalized preference over the meta-paths. So we further assign each user with a weight vector on meta-paths, representing user's personalized preference for each meta-path. It is more reasonable that each user has his/her personal interest preference in many real applications.

$$g(\{e_u^{(l)}\}) = \sum_{l=1}^{|\mathcal{P}|} w_u^{(l)}(\mathbf{M}^{(l)}e_u^{(l)} + b^{(l)}),\qquad(7)$$

where $w_u^{(l)}$ is the preference weight of user u over the lth meta-path.

- Personalized non-linear fusion. Linear fusion has limited expressive power in modeling complex data relations. Hence, we use non-linear function to enhance the fusion ability.

$$g(\{e_u^{(l)}\}) = \sigma\left(\sum_{l=1}^{|\mathcal{P}|} w_u^{(l)}\sigma(\mathbf{M}^{(l)}e_u^{(l)} + b^{(l)})\right),\qquad(8)$$

where $\sigma(\cdot)$ is a non-linear function, i.e., sigmoid function in our work. Although we only use two non-linear transformations, it is flexible to extend to multiple non-linear layers, e.g., Multi-Layer Perceptrons.

4.2.3 Model Learning

We blend the fusion function into matrix factorization framework for learning the parameters of the proposed model. The objective can be formulated as follows:

$$\pounds = \sum_{\langle u,i,r_{u,i}\rangle \in \mathcal{R}} (r_{u,i} - \widehat{r_{u,i}})^2 + \lambda \sum_u (\|\mathbf{x}_u\|_2 + \|\mathbf{y}_i\|_2$$
$$+ \|\gamma_u^{(U)}\|_2 + \|\gamma_i^{(I)}\|_2 + \|\Theta^{(U)}\|_2 + \|\Theta^{(I)}\|_2),\qquad(9)$$

where $\widehat{r_{u,i}}$ is the predicted rating using Eq. (5) by the proposed model, λ is the regularization parameter, and $\Theta^{(U)}$ and $\Theta^{(I)}$ are the parameters of the function $g(\cdot)$ for users and items respectively. We adopt SGD to efficiently optimize the final objective. The update of original latent factors $\{\mathbf{x}_u\}$ and $\{\mathbf{y}_i\}$ are the same as that of standard MF in Eq. (4). The parameters of the proposed model will be updated as follows:

$$\Theta_{u,l}^{(U)} \leftarrow \Theta_{u,l}^{(U)} - \eta \cdot \left(-\alpha(r_{u,i} - \widehat{r_{u,i}})\gamma_i^{(I)}\frac{\partial e_u^{(U)}}{\partial \Theta_{u,l}^{(U)}} + \lambda_\Theta \Theta_{u,l}^{(U)}\right),\quad(10)$$

$$\gamma_u^{(U)} \leftarrow \gamma_u^{(U)} - \eta \cdot (-\beta(r_{u,i} - \widehat{r_{u,i}})e_i^{(I)} + \lambda_\gamma \gamma_u^{(U)}),\qquad(11)$$

$$\Theta_{i,l}^{(I)} \leftarrow \Theta_{i,l}^{(I)} - \eta \cdot \left(-\beta(r_{u,i} - \widehat{r_{u,i}})\gamma_u^{(U)}\frac{\partial e_i^{(I)}}{\partial \Theta_{i,l}^{(I)}} + \lambda_\Theta \Theta_{i,l}^{(I)}\right),\quad(12)$$

$$\gamma_i^{(I)} \leftarrow \gamma_i^{(I)} - \eta \cdot (-\alpha(r_{u,i} - \widehat{r_{u,i}})e_u^{(U)} + \lambda_\gamma \gamma_i^{(I)}),\qquad(13)$$

where η is the learning rate, λ_Θ is the regularization for parameters $\Theta^{(U)}$ and $\Theta^{(I)}$, and λ_γ is the regularization for parameters $\gamma^{(U)}$ and $\gamma^{(I)}$. In our work, we utilize the sigmod function for non-linear transformation, and we can take advantage of the properties of sigmod function for ease of derivative calculation. It is worth noting that the symbol Θ denotes all the parameters in the fusion function, and the calculation of $\frac{\partial e_i}{\partial \Theta_{i,l}}$ will be different for different parameters in Θ. Next, we present the detailed derivation of $\frac{\partial e_i}{\partial \Theta_{i,l}}$ for

personalized non-linear fusion function

$$\frac{\partial e_i}{\partial \Theta_{i,l}} = \begin{cases} w_i^{(l)}\sigma(Z_s)\sigma(Z_f)(1-\sigma(Z_s))(1-\sigma(Z_f))e_i^{(l)}, & \Theta = M; \\ \\ w_i^{(l)}\sigma(Z_s)\sigma(Z_f)(1-\sigma(Z_s))(1-\sigma(Z_f)), & \Theta = b; \\ \\ \sigma(Z_s)\sigma(Z_f)(1-\sigma(Z_s)), & \Theta = w, \end{cases}\qquad(14)$$

where $Z_s = \sum_{l=1}^{|\mathcal{P}|} w_i^{(l)}\sigma(\mathbf{M}^{(l)}e_i^{(l)} + b^{(l)})$ and $Z_f = \mathbf{M}^{(l)}e_i^{(l)} + b^{(l)}$. The derivations of $\frac{\partial e_i}{\partial \Theta_{i,l}}$ can be calculated in the above way for both users and items. We omit the derivations for linear fusion functions, since it is relatively straightforward. The whole algorithm framework is shown in Algorithm 1. In line 1, we perform HIN embedding to obtain the representations of users and items. And in lines 3-16, we adopt the SGD algorithm to optimize the parameters in the fusion function and rating function.

Algorithm 1. The Overall Learning Algorithm of HERec

Input: the rating matrix $\mathcal{R}$; the learning rate η; the adjustable parameters α, β; the regularization parameter λ; the meta-path sets for users and items, $\mathcal{P}^{(U)}$ and $\mathcal{P}^{(I)}$.

Output: the latent factors for users and items, $\mathbf{x}$ and $\mathbf{y}$; the latent factors to pair HIN embedding of users and items, $\gamma^{(U)}$ and $\gamma^{(I)}$; the parameters of the fusion function for users and items, $\Theta^{(U)}$ and $\Theta^{(I)}$

1: Obtain the set of users' embeddings $\{e_u^{(l)}\}_{l=1}^{|\mathcal{P}^{(U)}|}$ and the set of items' embeddings $\{e_i^{(l)}\}_{l=1}^{|\mathcal{P}^{(I)}|}$

2: Initialize $\mathbf{x}, \mathbf{y}, \gamma^{(U)}, \gamma^{(I)}, \Theta^{(U)}, \Theta^{(I)}$ by standard normal distribution;

3: **while** not convergence **do**

4:    Randomly select a triple $\langle u,i,r_{u,i}\rangle \in \mathcal{R}$;

5:    Update $\mathbf{x}_u, \mathbf{y}_i$ by typical MF;

6:    **for** $l = 1$ to $|\mathcal{P}^{(U)}|$ **do**

7:       Calculate $\frac{\partial e_u^{(U)}}{\partial \Theta_{u,l}^{(U)}}$ by Eq. (14);

8:       Update $\Theta_{u,l}^{(U)}$ by Eq. (10);

9:    **end for**

10:   Update $\gamma_u^{(U)}$ by Eq. (11);

11:   **for** $l = 1$ to $|\mathcal{P}^{(I)}|$ **do**

12:      Calculate $\frac{\partial e_i^{(I)}}{\partial \Theta_{i,l}^{(I)}}$ by Eq. (14);

13:      Update $\Theta_{i,l}^{(I)}$ by Eq. (12);

14:   **end for**

15:   Update $\gamma_i^{(I)}$ by Eq. (13);

16: **end while**

17: **Return** $\mathbf{x}, \mathbf{y}, \gamma^{(U)}, \gamma^{(I)}, \Theta^{(U)}, \Theta^{(I)}$.

4.2.4 Complexity Analysis

HERec contains two major parts: (1) HIN embedding. The complexity of deepwalk is $O(\tau \cdot |\mathcal{V}| \cdot t \cdot w \cdot (d + d \cdot \log |\mathcal{V}|))$, where τ is the number of random walk, t is the length of random walk, w is the size of neighbor, d is the embedding dimension and $|\mathcal{V}|$ is the number of nodes in the network [63]. Hence, the total complexity of HIN embedding can be abbreviated as $O(|\mathcal{P}| \cdot \tau \cdot t \cdot w \cdot d \cdot (|\mathcal{U}| \cdot \log |\mathcal{U}| + |\mathcal{I}| \cdot \log |\mathcal{I}|))$ since the number of selected meta-paths is $|\mathcal{P}|$ and the number of users and items is $|\mathcal{U}|$ and $|\mathcal{I}|$, respectively. It

is worth noting that HIN embedding can be easily trained in parallel and we will implement it using a multi-thread mode in order to improve the efficiency of model. (2) Matrix factorization. For each triplet $\langle u, i, r_{u,i} \rangle$, updating x_u, y_i, $\gamma_u^{(U)}$, $\gamma_i^{(I)}$ takes $\mathcal{O}(D)$ time, where D is the number of latent factors. And updating $\mathbf{\Theta}_u^{(U)}$, $\mathbf{\Theta}_i^{(I)}$ takes $\mathcal{O}(|\mathcal{P}| \cdot D \cdot d)$ time to learn the transformation matrices $\mathbf{M}$ for all meta-paths. In the proposed approach, $|\mathcal{P}|$ is generally small, and d and D are at most several hundreds, which makes the proposed method efficient in large datasets.[1] Specially, SGD has very good practice performance, and we have found that it has a fast convergence rate on our datasets. Concretely, about 40 to 60 iterations are required for dense datasets (i.e., Douban Movie and Book), while about 20 iterations are required for sparse datasets (i.e., Yelp).

5   EXPERIMENTS

In this section, we will demonstrate the effectiveness of HERec by performing experiments on three real datasets compared to the state-of-the-art recommendation methods.

5.1   Evaluation Datasets

We adopt three widely used datasets from different domains, consisting of Douban Movie dataset[2] from movie domain, Douban Book dataset[3] from book domain, and Yelp dataset[4] from business domain. Douban Movie dataset includes 13,367 users and 12,677 movies with 1,068,278 movie ratings ranging from 1 to 5. Moreover, the dataset also includes social relations and attribute information of users and movies. As for the Douban Book dataset, it includes 13,024 users and 22,347 books with 792,026 ratings ranging from 1 to 5. Yelp dataset records user ratings on local businesses and contains social relations and attribute information of businesses, which includes 16,239 users and 14,282 local businesses with 198,397 ratings raging from 1 to 5. The detailed descriptions of the three datasets are shown in Table 2, and the network schemas of the three datasets are plotted in Fig. 2. Besides the domain variation, these three datasets also have different rating sparsity degrees: the Yelp dataset is very sparse, while the Douban Movie dataset is much denser.

5.2   Metrics

We use the widely used mean absolute error (MAE) and root mean square error (RMSE) to measure the quality of recommendation of different models. The metrics MAE and RMSE are defined as follows:

$$MAE = \frac{1}{|\mathcal{D}_{test}|} \sum_{(i,j) \in \mathcal{D}_{test}} |r_{i,j} - \widehat{r_{i,j}}|, \qquad (15)$$

$$RMSE = \sqrt{\frac{1}{|\mathcal{D}_{test}|} \sum_{(i,j) \in \mathcal{D}_{test}} (r_{i,j} - \widehat{r_{i,j}})^2}, \qquad (16)$$

where $r_{i,j}$ is the actual rating that user i assigns to item j, $\widehat{r_{i,j}}$ is the predicted rating from a model, and $\mathcal{D}_{test}$ denotes the test set of rating records. From the definition, we can see that a smaller MAE or RMSE value indicates a better performance.

5.3   Methods to Compare

We consider the following methods to compare:
- *PMF* [62]: It is the classic probabilistic matrix factorization model by explicitly factorizing the rating matrix into two low-dimensional matrices.
- *SoMF* [19]: It incorporates social network information into the recommendation model. The social relations are characterized by a social regularization term, and integrated into the basic matrix factorization model.
- FM_{HIN}: It is the context-aware factorization machine [64], which is able to utilize various kinds of auxiliary information. In our experiments, we extract heterogeneous information as context features and incorporate them into the factorization machine for rating prediction. We implement the model with the code from the authors.[5]
- *HeteMF* [7]: It is a MF based recommendation method, which utilizes entity similarities calculated over HINs using meta-path based algorithms.
- *SemRec* [10]: It is a collaborative filtering method on weighted heterogeneous information network, which is constructed by connecting users and items with the same ratings. It flexibly integrates heterogeneous information for recommendation by weighted meta-paths and a weight ensemble method. We implement the model with the code from the authors.[6]
- *DSR* [12]: It is a MF based recommendation method with dual similarity regularization, which imposes the constraint on users and items with high and low similarities simultaneously.
- $HERec_{dw}$: It is a variant of HERec, which adopts the homogeneous network embedding method deepwalk [13] and ignores the heterogeneity of nodes in HINs. We use the code of deepwalk provided by the authors.[7]
- $HERec_{mp}$: It can be viewed as a variant of HERec, which combines the heterogeneous network embedding of meta-path2vec++ [58] and still preserves the superiority of HINs. We use the code of metapath2vec++ provided by the authors.[8]
- *HERec*: It is the proposed recommendation model based on heterogeneous information network embedding. In the effectiveness experiments, we utilize the personalized non-linear fusion function, which is formulated as Eq. (8). And the performance of different fusion function will be reported in the later section.

The selected baselines have a comprehensive coverage of existing rating prediction methods. PMF and SoMF are classic MF based rating prediction methods; FM_{HIN}, HeteMF, SemRec and DSR are HIN based recommendation methods using meta-path based similarities, which represent state-of-the-art HIN based methods. The proposed approach contains an HIN embedding component, which can be replaced by other network embedding methods. Hence, two variants $HERec_{dw}$ and $HERec_{mp}$ are adopted as baselines.

Among these methods, HIN based methods need to specify the used meta-paths. We present the used meta-paths in

---

Table 2. Following [4], we only select short meta-paths of at most four steps, since long meta-paths are likely to introduce noisy semantics. We set parameters for HERec as follows: the embedding dimension number $d = 64$ and the tuning coefficients $\alpha = \beta = 1.0$. Following [13], the path length for random walk is set to 40. For HERec$_{dw}$ and HERec$_{mp}$, we set the embedding dimension number $d = 64$ for fairness. Following [11], [12], we set the number of latent factors for all the MF based methods to 10. Other baseline parameters either adopt the original optimal setting or are optimized by using a validation set of 10 percent training data.

5.4 Effectiveness Experiments

For each dataset, we split the entire rating records into a training set and a test set. For Double Movie and Book datasets, we set four training ratios as in $\{80\%, 60\%, 40\%, 20\%\}$; while for Yelp dataset, following [10], we set four larger training ratios as in $\{90\%, 80\%, 70\%, 60\%\}$, since it is too sparse. For each ratio, we randomly generate ten evaluation sets. We average the results as the final performance. The results are shown in Table 3. The major findings from the experimental results are summarized as follows:

(1) Among these baselines, HIN based methods (HeteMF, SemRec, FM$_{HIN}$ and DSR) perform better than traditional MF based methods (PMF and SoMF), which indicates the usefulness of the heterogeneous information. It is worthwhile to note that the FM$_{HIN}$ model works very well among the three HIN based baselines. An intuitive explanation is that in our datasets, most of the original features are the attribute information of users or items, which are likely to contain useful evidence to improve the recommendation performance.

(2) The proposed HERec method (corresponding to the last column) is consistently better than the baselines, ranging from PMF to DSR. Compared with other HIN based methods, HERec adopts a more principled way to leverage HINs for improving recommender systems, which provides better information extraction (a new HIN embedding model) and utilization (an extended MF model). Moreover, the superiority of the proposed HERec becomes more significant with less training data. In particular, the improvement ratio of HERec over the PMF is up to 40 with 20 percent training data on the Douban Book dataset, which indicates a significant performance boost. As mentioned previously, the Yelp dataset is very sparse. In this case, even with 60 percent training data, the HERec model improves over PMF by about 26 percent. As a comparison, with the same training ratio (i.e., 60 percent), the improvement ratios over PMF are about 29 percent in terms of RMSE scores. These results indicate the effectiveness of the proposed approach, especially in a sparse dataset.

(3) Considering the two HERec variants HERec$_{dw}$ and HERec$_{mp}$, it is easy to see HERec$_{dw}$ performs much worse than HERec$_{mp}$. The major difference lies in the network embedding component. HERec$_{mp}$ adopts the recently proposed HIN embedding method *metapath2vec* [58], while HERec$_{dw}$ ignores data heterogeneity and casts HINs as homogeneous networks for learning. It indicates HIN embedding methods are important to HIN based recommendation. The major difference between the proposed embedding method and metapath2vec (adopted by HERec$_{mp}$) lies in that we try to transform heterogeneous network information into homogeneous neighborhood, while metapath2vec tries to map all types of entities into the same

representation space using heterogeneous sequences. Indeed, metapath2vec is a general-purpose HIN embedding method, while the proposed embedding method aims to improve the recommendation performance. Our focus is to learn the embeddings for users and items, while objects of other types are only used as the bridge to construct the homogeneous neighborhood. Based on the results (HERec $>$ HERec$_{mp}$), we argue that it is more effective to perform task-specific HIN embedding for improving the recommendation performance.

5.5 Detailed Analysis of The Proposed Approach

In this part, we perform a series of detailed analysis for the proposed approach.

5.5.1 Selection of Different Fusion Functions

HERec requires a principled fusion way to transform node embeddings into a more suitable form that is useful to enhance recommendation performance. Therefore, we will discuss the impact of different fusion functions on the recommendation performance. For convenience, we call the HERec variant with the simple linear fusion function (Eq. (6)) as HERec$_{sl}$, the variant with personalized linear fusion function (Eq. (7)) as HERec$_{pl}$, and the variant with personalized non-linear fusion function (Eq. (8)) as HERec$_{pnl}$. We present the performance comparison of the three variants of HERec on the three datasets in Fig. 4.

As shown in Fig. 4, we can find that overall performance ranking is as follows: HERec$_{pnl}$ $>$ HERec$_{pl}$ $>$ HERec$_{sl}$. Among the three variants, the simple linear fusion function performs the worst, as it ignores the personalization and non-linearity. Indeed, users are likely to have varying preferences over meta-paths [10], which should be considered in meta-path based methods. The personalization factor improves the performance significantly. As a comparison, the performance improvement of HERec$_{pnl}$ over HERec$_{pl}$ is relatively small. A possible reason is that the incorporation of personalized combination parameters increases the capability of linear models. Nonetheless, HERec$_{pnl}$ still performs the best by considering personalization and non-linear transformation. In a complicated recommender setting, HIN embeddings may not be directly applicable in recommender systems, where a non-linear mapping function is preferred.

Since HERec$_{pnl}$ is the best variant of the proposed model, in what follows, HERec will use the personalized non-linear fusion function, i.e., HERec$_{pnl}$ by default.

5.5.2 Cold-Start Prediction

HINs are particularly useful to improve cold-start prediction, where there are fewer rating records but heterogeneous context information is available. We consider studying the performance *w.r.t.* different cold-start degrees, i.e., the rating sparsity. To test it, we first categorize "cold" users into three groups according to the numbers of their rating records, i.e., $(0, 5]$, $(5, 15]$ and $(15, 30]$. It is easy to see that the case for the first group is the most difficult, since users from this group have fewest rating records. Here, we only select the baselines which use HIN based information for recommendation, including SoMF, HeteMF, SemRec, DSR and FM$_{HIN}$. We present the performance comparison of different methods in Fig. 5. For convenience, we report the improvement ratios *w.*

TABLE 3
Results of Effectiveness Experiments on Three Datasets

Dataset	Training	Metrics	PMF	SoMF	FM$_{HIN}$	HeteMF	SemRec	DSR	HERec$_{dw}$	HERec$_{mp}$	HERec
Douban Movie	80%	MAE Improve	0.5741	0.5817 −1.32%	0.5696 +0.78%	0.5750 −0.16%	0.5695 +0.80%	0.5681 +1.04%	0.5703 +0.66%	**0.5515** +3.93%	0.5519 +3.86%
		RMSE Improve	0.7641	0.7680 −0.07%	0.7248 +5.55%	0.7556 +1.53%	0.7399 +3.58%	0.7225 +5.85%	0.7446 +2.97%	0.7121 +7.20%	**0.7053** +8.09%
	60%	MAE Improve	0.5867	0.5991 −2.11%	0.5769 +1.67%	0.5894 −0.46%	0.5738 +2.19%	0.5831 +0.61%	0.5838 +0.49%	0.5611 +4.36%	**0.5587** +4.77%
		RMSE Improve	0.7891	0.7950 −0.75%	0.7842 +0.62%	0.7785 +1.34%	0.7551 +4.30%	0.7408 +6.12%	0.7670 +2.80%	0.7264 +7.94%	**0.7148** +9.41%
	40%	MAE Improve	0.6078	0.6328 −4.11%	0.5871 +3.40%	0.6165 −1.43%	0.5945 +2.18%	0.6170 −1.51%	0.6073 +0.08%	0.5747 +5.44%	**0.5699** +6.23%
		RMSE Improve	0.8321	0.8479 −1.89%	0.7563 +9.10%	0.8221 +1.20%	0.7836 +5.82%	0.7850 +5.66%	0.8057 +3.17%	0.7429 +10.71%	**0.7315** +12.09%
	20%	MAE Improve	0.7247	0.6979 +3.69%	0.6080 +16.10%	0.6896 +4.84%	0.6392 +11.79%	0.6584 +9.14%	0.6699 +7.56%	0.6063 +16.33%	**0.5900** +18.59%
		RMSE Improve	0.9440	0.9852 −4.36%	0.7878 +16.55%	0.9357 +0.88%	0.8599 +8.91%	0.8345 +11.60%	0.9076 +3.86%	0.7877 +16.56%	**0.7660** +18.86%
Douban Book	80%	MAE Improve	0.5774	0.5756 +0.31%	0.5716 +1.00%	0.5740 +0.59%	0.5675 +1.71%	0.5740 +0.59%	0.5875 −1.75%	0.5591 +3.17%	**0.5502** +4.71%
		RMSE Improve	0.7414	0.7302 +1.55%	0.7199 +2.94%	0.7360 +0.77%	0.7283 +1.81%	0.7206 +2.84%	0.7450 −0.44%	0.7081 +4.53%	**0.6811** +8.17%
	60%	MAE Improve	0.6065	0.5903 +2.67%	0.5812 +4.17%	0.5823 +3.99%	0.5833 +3.83%	0.6020 +0.74%	0.6203 −2.28%	0.5666 +6.58%	**0.5600** +7.67%
		RMSE Improve	0.7908	0.7518 +4.93%	0.7319 +7.45%	0.7466 +5.59%	0.7505 +5.10%	0.7552 +4.50%	0.7905 +0.04%	0.7318 +7.46%	**0.7123** +9.93%
	40%	MAE Improve	0.6800	0.6161 +9.40%	0.6028 +11.35%	0.5982 +12.03%	0.6025 +11.40%	0.6271 +7.78%	0.6976 −2.59%	0.5954 +12.44%	**0.5774** +15.09%
		RMSE Improve	0.9203	0.7936 +13.77%	0.7617 +17.23%	0.7779 +15.47%	0.7751 +15.78%	0.7730 +16.01%	0.9022 +1.97%	0.7703 +16.30%	**0.7400** +19.59%
	20%	MAE Improve	1.0344	0.6327 +38.83%	0.6396 +38.17%	0.6311 +38.99%	0.6481 +37.35%	0.6300 +39.10%	1.0166 +1.72%	0.6785 +34.41%	**0.6450** +37.65%
		RMSE Improve	1.4414	0.8236 +42.86%	0.8188 +43.19%	0.8304 +42.39%	0.8350 +42.07%	0.8200 +43.11%	1.3205 +8.39%	0.8869 +38.47%	**0.8581** +40.47%
Yelp	90%	MAE Improve	1.0412	1.0095 +3.04%	0.9013 +13.44%	0.9487 +8.88%	0.9043 +13.15%	0.9054 +13.04%	1.0388 +0.23%	0.8822 +15.27%	**0.8395** +19.37%
		RMSE Improve	1.4268	1.3392 +6.14%	1.1417 +19.98%	1.2549 +12.05%	1.1637 +18.44%	1.1186 +21.60%	1.3581 +4.81%	1.1309 +20.74%	**1.0907** +23.56%
	80%	MAE Improve	1.0791	1.0373 +3.87%	0.9038 +16.25%	0.9654 +10.54%	0.9176 +14.97%	0.9098 +15.69%	1.0750 +0.38%	0.8953 +17.03%	**0.8475** +21.46%
		RMSE Improve	1.4816	1.3782 +6.98%	1.1497 +22.40%	1.2799 +13.61%	1.1771 +20.55%	1.1208 +24.35%	1.4075 +5.00%	1.1516 +22.27%	**1.1117** +24.97%
	70%	MAE Improve	1.1170	1.0694 +4.26%	0.9108 +18.46%	0.9975 +10.70%	0.9407 +15.78%	0.9429 +15.59%	1.1196 −0.23%	0.9043 +19.04%	**0.8580** +23.19%
		RMSE Improve	1.5387	1.4201 +7.71%	1.1651 +24.28%	1.3229 +14.02%	1.2108 +21.31%	1.1582 +24.73%	1.4632 +4.91%	1.1639 +24.36%	**1.1256** +26.85%
	60%	MAE Improve	1.1778	1.1135 +5.46%	0.9435 +19.89%	1.0368 +11.97%	0.9637 +18.18%	1.0043 +14.73%	1.1691 +0.74%	0.9257 +21.40%	**0.8759** +25.63%
		RMSE Improve	1.6167	1.4748 +8.78%	1.2039 +25.53%	1.3713 +15.18%	1.2380 +23.42%	1.2257 +24.19%	1.5182 +6.09%	1.1887 +26.47%	**1.1488** +28.94%

A smaller MAE or RMSE value indicates a better performance. For ease of reading the results, we also report the improvement w.r.t. the PMF model for each other method. A larger improvement ratio indicates a better performance.

r.t. PMF. Overall, all the comparison methods are better than PMF (i.e., a positive y-axis value). The proposed method performs the best among all the methods, and the improvement over PMF becomes more significant for users with fewer rating records. The results indicate that HIN based information is effective to improve the recommendation performance, and the proposed HERec method can effectively utilize HIN information in a more principled way.
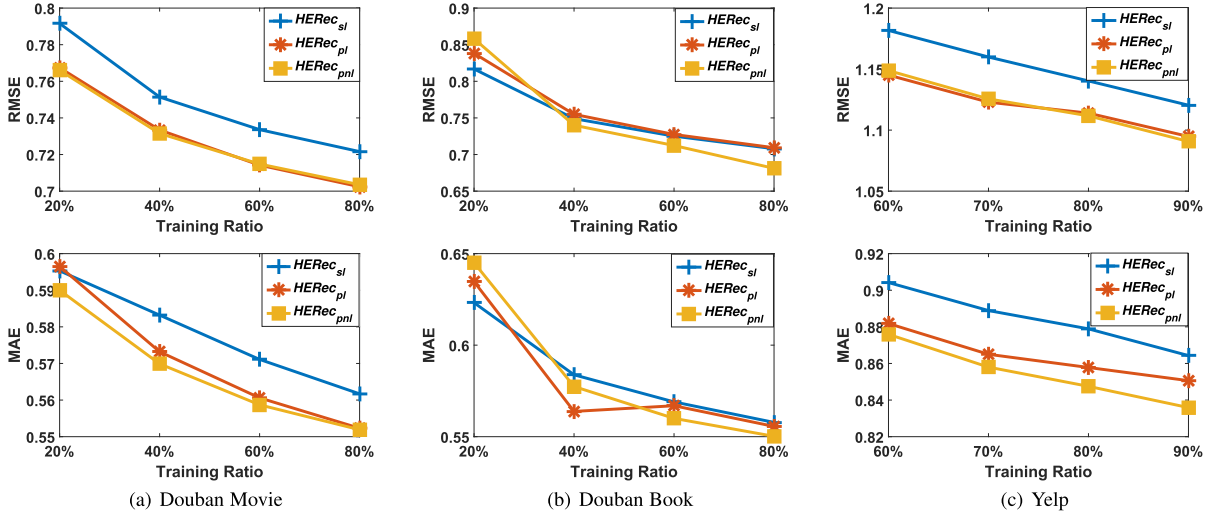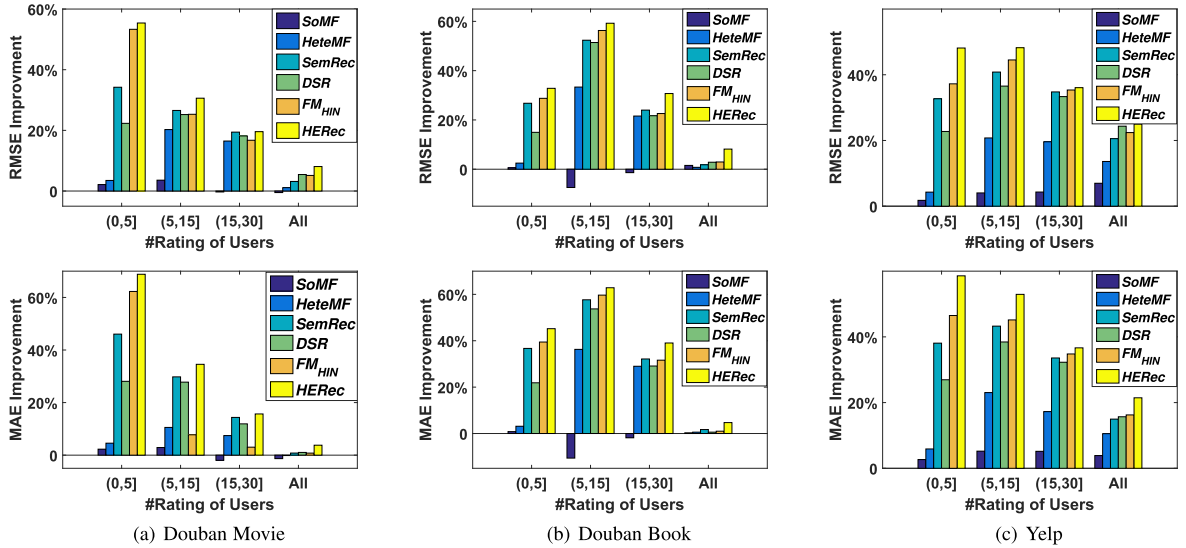
Fig. 4. Performance comparison of different fusion functions on three datasets.

Fig. 5. Performance comparison of different methods for <mark>cold-start prediction</mark> on three datasets. y-axis denotes the improvement ratio over PMF.

5.5.3 Impact of Different Meta-Paths

As shown in Table 2, the proposed approach uses a selected set of meta-paths. To further analyze the impact of different meta-paths, we gradually incorporate these meta-paths into the proposed approach and check the performance change. In Fig. 6, we can observe that generally the performance improves (i.e., becoming smaller) with the incorporation of more meta-paths. Both meta-paths starting with the user type and item type are useful to improve the performance. However, it does not always yield the improvement with more meta-paths, and the performance slightly fluctuates. The reason is that some meta-paths may contain noisy or conflict information with existing ones. Another useful observation is that the performance quickly achieves a relatively good performance with the incorporation of only a few meta-paths. This confirms previous finding [10]: a small number of high-quality meta-paths are able to lead to large performance improvement. Hence, as mentioned before, we can effectively control the model complexity by selecting just a few meta-paths.

5.5.4 Parameter Tuning

We first study the impact of the dimension of embeddings for the proposed model. We vary embedding dimensions in the set of $\{16, 32, 64, 128, 256\}$ and select $HERec_{mp}$ as the reference baseline. As shown in Fig. 7, our method consistently outperforms $HERec_{mp}$ and achieves the optimal performance when the dimension of embeddings is around 64. Moreover, the performance of our method is more stable when varying the dimension of embeddings compared with $HERec_{mp}$.

For matrix factorization based methods, an important parameter to tune is the number of latent factors. The proposed model also involves such a parameter. We vary it from 5 to 40 with a step of 5, and examine how the performance changes $w.r.t$ the number of latent factors. We present the tuning results in Fig. 8. As we can see, using 10 latent factors yields the best performance, indicating that the number of latent factors should be set to a small number.

Next, we fix the number of latent factors as ten, and tune two other parameters α and β (Eq. (5)), which are used to integrate different terms as weights. Now we examine how they influence the model performance. For both parameters, we vary them in the set of $\{0.1, 0.5, 1, 2\}$. As shown in Fig. 9, the optimal performance is obtained near $(1, 1)$, i.e., both α and β are around 1. The results show that the HIN embeddings from both the user and item sides are important to improve the prediction performance. Overall, the change
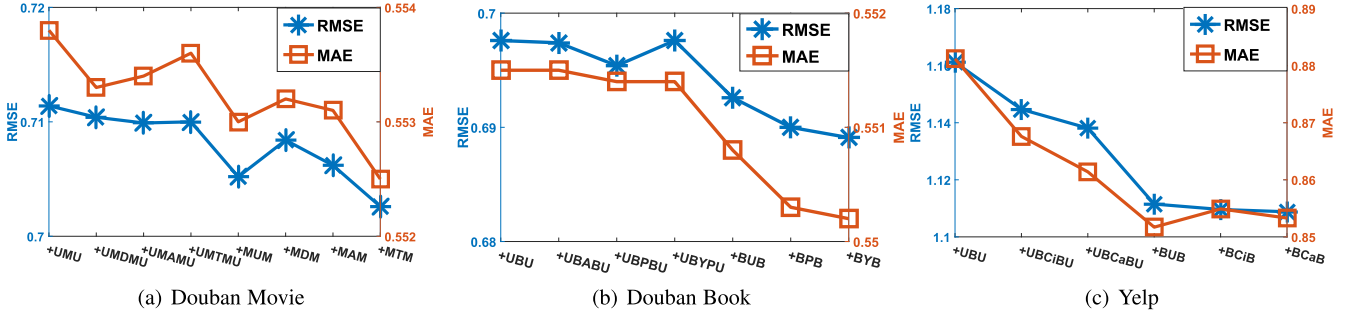
Fig. 6. Performance change of HERec when gradually incorporating meta-paths.
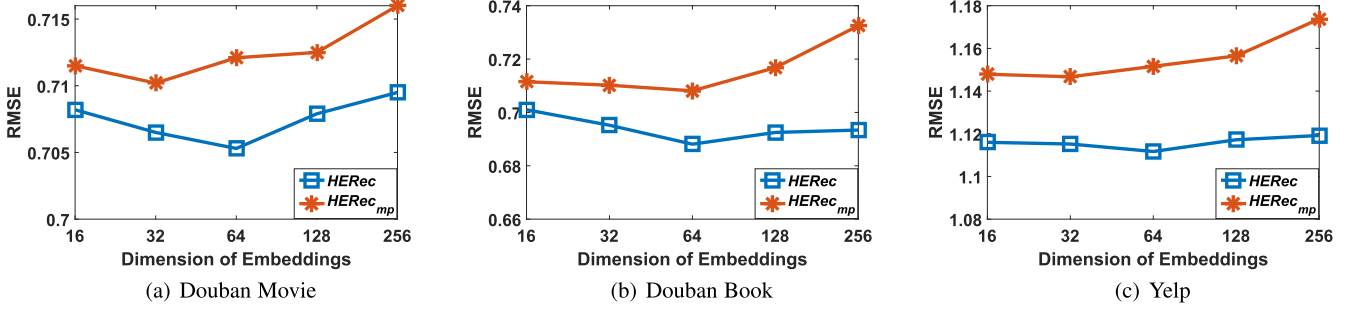

Fig. 7. Performance change with respect to the dimension of embeddings on three datasets.
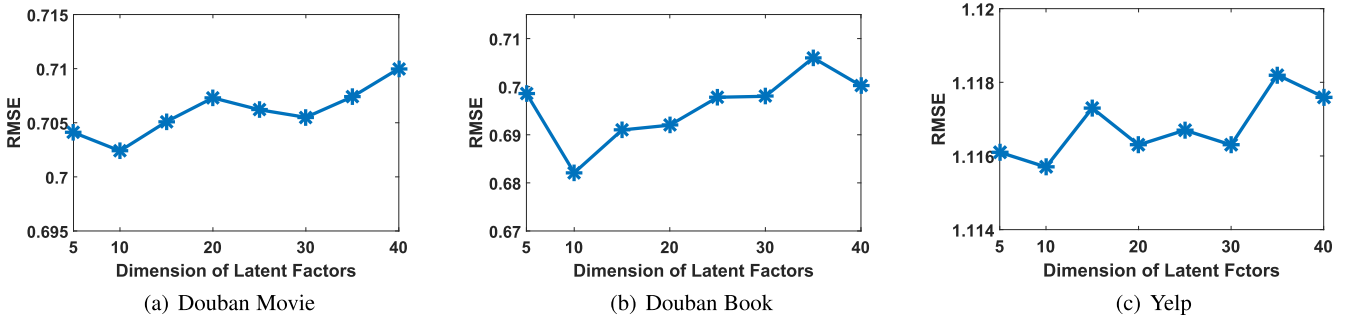

Fig. 8. Performance change with respect to the dimension of latent factors on three datasets.
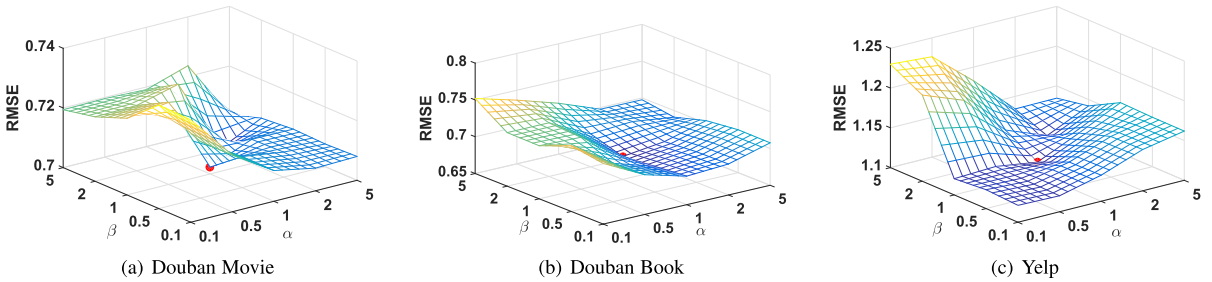

Fig. 9. Varying parameters α and β on the three datasets.

trend is smooth, indicating that the proposed model is not very sensitive to the two parameters.

6 CONCLUSION

In this paper, we proposed a novel heterogeneous information network embedding based approach (i.e., HERec) to effectively utilizing auxiliary information in HINs for recommendation. We designed a new random walk strategy based on meta-paths to derive more meaningful node sequences for network embedding. Since embeddings based on different meta-paths contain different semantic, the learned embeddings were further integrated into an extended matrix factorization

model using a set of fusion functions. Finally, the extended matrix factorization model together with fusion functions were jointly optimized for the rating prediction task. HERec aimed to learn useful information representations from HINs guided by the specific recommendation task, which distinguished the proposed approach from existing HIN based recommendation methods. Extensive experiments on three real datasets demonstrated the effectiveness of HERec. We also verified the ability of HERec to alleviate cold-start problem and examine the impact of meta-paths on performance.

As future work, we will investigate into how to apply deep learning methods (e.g., convolutional neural networks,

auto encoder) to better fuse the embeddings of multiple meta-paths. In addition, we only use the meta-paths which have the same starting and ending types to effectively extract network structure features in this work. Therefore, it is interesting and natural to extend the proposed model to learn the embeddings of any nodes with arbitrary meta-paths. As a major issue of recommender systems, we will also consider how to enhance the explainablity of the recommendation method based on the semantics of meta-paths.

ACKNOWLEDGMENTS

REFERENCES

[1] M. B. Dias, D. Locher, M. Li, W. El-Deredy , and P. J. Lisboa, "The value of personalised recommender systems to e-business: A case study," in *Proc. 2nd ACM Conf. Recommender Syst.*, 2008, pp. 291–294.

[2] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Proc. Recommender Syst. Handbook*, 2015, pp. 77–118.

[3] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, in *The Adaptive Web*. Berlin, Germany: Springer, 2007, pp. 291–324.

[4] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proc. 37th Very Large Data Bases*, vol. 4, no. 11, pp. 992–1003, 2011.

[5] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 17–37, Jan. 2017.

[6] C. Shi and P. S. Yu, "Heterogeneous information network analysis and applications," in *Data Analytics*. New York, NY, USA: Springer, 2017, pp. 1–227.

[7] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han, "Collaborative filtering with entity similarity regularization in heterogeneous information networks," in *Proc. 1st Int. Joint Conf. Artif. Intell. Workshop Heterogeneous Inf. Netw. Anal.*, 2013.

[8] W. Feng and J. Wang, "Incorporating heterogeneous information for personalized tag recommendation in social tagging systems," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1276–1284.

[9] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 283–292.

[10] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 453–462.

[11] C. Shi, J. Liu, F. Zhuang, P. S. Yu, and B. Wu, "Integrating heterogeneous information via flexible regularization framework for recommendation," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 835–859, 2016.

[12] J. Zheng, J. Liu, C. Shi, F. Zhuang, J. Li, and B. Wu, "Recommendation in heterogeneous information network via dual similarity regularization," *Int. J. Data Sci. Analytics*, vol. 3, no. 1, pp. 35–48, 2017.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.

[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Comput.*, vol. 42, no. 8, pp. 30–37, 2009.

[16] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic, "Adaptive diversification of recommendation results via latent factor portfolio," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2012, pp. 175–184.

[17] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars: a location-content-aware recommender system," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 221–229.

[18] L. Hong, A. S. Doumith, and B. D. Davison, "Co-factorization machines: Modeling user interests and predicting individual decisions in twitter," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 557–566.

[19] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.

[20] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 105–112.

[21] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 325–334.

[22] S. Sedhain, A. K. Menon, S. Sanner, L. Xie, and D. Braziunas, "Low-rank linear cold-start recommendation from social data." in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1502–1508.

[23] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," in *Proc. 10th IEEE Int. Conf. Data Mining Series*, 2010, pp. 176–185.

[24] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 173–182.

[25] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen, "Social collaborative filtering for cold-start recommendations," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 345–348.

[26] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 425–434.

[27] R. He and J. McAuley, "Vbpr: Visual bayesian personalized ranking from implicit feedback," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 144–150.

[28] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 353–362.

[29] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu, "Svdfeature: A toolkit for feature-based collaborative filtering," *J. Mach. Learn. Res.*, vol. 13, pp. 3619–3622, 2012.

[30] S. Rendle, "Factorization machines," in *Proc. 10th IEEE Int. Conf. Data Mining Series*, 2010, pp. 995–1000.

[31] Y. Sun and J. Han, "Mining heterogeneous information networks: A structural analysis approach," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[32] M. Ou, P. Cui, F. Wang, J. Wang, W. Zhu, and S. Yang, "Comparing apples to oranges: A scalable solution with heterogeneous hashing," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 230–238.

[33] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Mach. Learn.*, vol. 81, no. 1, pp. 53–67, 2010.

[34] C. Shi, X. Kong, Y. Huang, P. S. Yu, and B. Wu, "Hetesim: A general framework for relevance measure in heterogeneous networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2479–2492, Oct. 2014.

[35] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, "Recommendation in heterogeneous information networks with implicit user feedback," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 347–350.

[36] C. Luo, W. Pang, Z. Wang, and C. Lin, "Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations," in *Proc. 14th IEEE Int. Conf. Data Mining Series*, 2014, pp. 917–922.

[37] J. Zheng, J. Liu, C. Shi, F. Zhuang, J. Li, and B. Wu, "Dual similarity regularization for recommendation," in *Proc. 20th Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2016, pp. 542–554.

[38] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 635–644.

[39] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *J. Amer. Statistical Assoc.*, vol. 97, no. 460, pp. 1090–1098, 2002.

[40] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[41] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," arXiv: 1711.08752, 2017.

[42] P. Cui, S. Liu, and W. Zhu, "General knowledge embedded image representation learning," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 198–207, Jan. 2018.

[43] C. Tu, W. Zhang, Z. Liu, and M. Sun, "Max-margin deepwalk: Discriminative learning of network representation." in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 3889–3895.

[44] X. Wei, L. Xu, B. Cao, and P. S. Yu, "Cross view link prediction by learning noise-resilient representation consensus," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1611–1619.

[45] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1145–1152.

[46] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 59–66.

[47] Z. Sun, J. Yang, J. Zhang, A. Bozzon, Y. Chen, and C. Xu, "Mrlr: Multi-level representation learning for personalized ranking in recommendation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2807–2813.

[48] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[49] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1225–1234.

[50] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 891–900.

[51] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," *Netw.*, vol. 11, no. 9, 2016, Art. no. 12.

[52] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 2111–2117.

[53] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Homophily, structure, and content augmented network representation learning," in *Proc. 16th IEEE Int. Conf. Data Mining Series*, 2016, pp. 609–618.

[54] S. Chang, W. Han, J. Tang, G. J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 119–128.

[55] J. Tang, M. Qu, and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1165–1174.

[56] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Embedding of embedding (EOE): Joint embedding for coupled heterogeneous networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 741–749.

[57] T. Chen and Y. Sun, "Task-guided and path-augmented heterogeneous network embedding for author identification," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 295–304.

[58] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 135–144.

[59] T. Fu, W. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1797–1806.

[60] Y. Sun and J. Han, "Mining heterogeneous information networks: Principles and methodologies," *Synthesis Lectures Data Mining Knowl. Discovery*, pp. 1–159, 2012.

[61] Y. Sun, Y. Yu, and J. Han, "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 797–806.

[62] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[63] H. Chen, B. Perozzi, Y. Hu, and S. Skiena, "Harp: Hierarchical representation learning for networks," in *Proc. 32th AAAI Conf. Artif. Intell.*, 2018, pp. 2127–2134.

[64] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, 2012, Art, no. 57.

Chuan Shi received the BS degree from Jilin University, in 2001, the MS degree from the Wuhan University, in 2004, and the PhD degree from the ICT of Chinese Academic of Sciences, in 2007. He joined the Beijing University of Posts and Telecommunications as a lecturer, in 2007, and is a professor and deputy director of the Beijing Key Lab of Intelligent Telecommunications Software and Multimedia at present. His research interests include data mining, machine learning, and evolutionary computing. He has published more than 60 papers in refereed journals and conferences, such as SIGKDD, IJCAI, CIKM, the *IEEE Transactions on Knowledge and Data Engineering*, the *World Wide Web Journal*, and the *ACM Transactions on Intelligent Systems and Technology*. He is a member of the IEEE.

Binbin Hu received the BS degree from the Beijing University of Posts and Telecommunications (BUPT), in 2016. He is currently working toward the master's degree at BUPT. His research interests include data mining and machine learning.

Wayne Xin Zhao received the PhD degree from Peking University, in 2014. He is currently an associate professor with the School of Information, Renmin University of China. His research interests include web text mining and natural language processing. He has published more than 50 papers in refereed conferences and journals such as ACL, EMNLP, COLING, CIKM, SIGIR, SIGKDD, AAAI, the *ACM Transactions on Information Systems*, the *ACM Transactions on Knowledge Discovery from Data*, the *ACM Transactions on Intelligent Systems and Technology* and the *IEEE Transactions on Knowledge and Data Engineering*. He is a member of the IEEE.

Philip S. Yu received the BS degree in electrical engineering from National Taiwan University, the MS and PhD degrees in electrical engineering from Stanford University, and the MBA degree from New York University. He is a distinguished professor in computer science with the University of Illinois at Chicago and also holds the Wexler chair in information technology. His research interests include big data, including data mining, data stream, database, and privacy. He has published more than 1,000 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. He is the recipient of the ACM SIGKDD 2016 Innovation Award for his influential research and scientific contributions on mining, fusion, and anonymization of big data, the IEEE Computer Societys 2013 Technical Achievement Award for pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining, and anonymization of big data, and the Research Contributions Award from IEEE ICDM in 2003 for his pioneering contributions to the field of data mining. He also received the ICDM 2013 10-year Highest-Impact Paper Award, and the EDBT Test of Time Award (2014). He was the editor-in-chiefs of the *ACM Transactions on Knowledge Discovery from Data* (2011-2017) and the *IEEE Transactions on Knowledge and Data Engineering* (2001-2004). He is a fellow of the IEEE and ACM.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.