

An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph

Jiarui Jin¹, Jiarui Qin¹, Yuchen Fang¹, Kounianhua Du¹, Weinan Zhang¹, Yong Yu¹,

Zheng Zhang², Alexander J. Smola².

¹Shanghai Jiao Tong University, ²Amazon Web Services

{jinjiaru97, qjr1996, arthur_fyc, 774581965, wnzhang, yyu}@sjtu.edu.cn, zhaz@amazon.com, alex@smola.org

ABSTRACT

There is an influx of heterogeneous information network (HIN) based recommender systems in recent years since HIN is capable of characterizing complex graphs and contains rich semantics. Although the existing approaches have achieved performance improvement, while practical, they still face the following problems. On one hand, most existing HIN-based methods rely on explicit path reachability to leverage path-based semantic relatedness between users and items, e.g., metapath-based similarities. These methods are hard to use and integrate since path connections are sparse or noisy, and are often of different lengths. On the other hand, other graph-based methods aim to learn effective heterogeneous network representations by compressing node together with its neighborhood information into single embedding before prediction. This weakly coupled manner in modeling overlooks the rich interactions among nodes, which introduces an early summarization issue. In this paper, we propose an end-to-end Neighborhood-based Interaction Model for Recommendation (NIRec) to address above problems. Specifically, we first analyze the significance of learning interactions in HINs and then propose a novel formulation to capture the interactive patterns between each pair of nodes through their metapath-guided neighborhoods. Then, to explore complex interactions between metapaths and deal with the learning complexity on large-scale networks, we formulate interaction in a convolutional way and learn efficiently with fast Fourier transform. The extensive experiments on four different types of heterogeneous graphs demonstrate the performance gains of NIRec comparing with state-of-the-arts. To the best of our knowledge, this is the first work providing an efficient neighborhood-based interaction model in the HIN-based recommendations.

CCS CONCEPTS

• Information systems → Data mining; • Computer systems organization → Heterogeneous (hybrid) systems.

KEYWORDS

Recommender System; Neighborhood-based Interaction; Heterogeneous Information Network

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7998-4/20/08...\$15.00

<https://doi.org/10.1145/3394486.3403050>

ACM Reference Format:

Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, Alexander J. Smola. 2020. An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA* ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3394486.3403050>

1 INTRODUCTION

During the recent decade, the techniques of recommender systems have developed from pure collaborative filtering based on only user-item interactions [12] to deep neural networks with various kinds of auxiliary data containing complex and useful information [6, 19].

Recently, the heterogeneous information network (HIN)¹, consisting of multiple types of nodes and/or links, has been leveraged as a powerful modeling method to fuse complex information and successfully applied to many recommender system tasks, which are called HIN-based recommendation methods [22, 24]. In Figure 1, we present an instance of movie data characterized by an HIN. We can easily see that the HIN contains multiple types of entities connected by different types of relations. A variety of graph representation learning methods have been proposed on HINs to capture the rich semantic information, which roughly fall into the following two categories.

One school is graph-based methodologies such as HetGNN [35], where deep neural network architectures such as GCN [11] on homogeneous graphs are extended to enhance aggregating feature information of neighboring nodes on heterogeneous graphs. However, these technologies usually compress the information of a node and its neighborhood into single embedding vector before making prediction [4, 22]. In this case, only two nodes and one edge are activated, yet other nodes and their connections are mixed and relayed, which introduces an **early summarization** issue [18]. The other school is metapath-based approaches. A metapath means composite relation connecting two objects in network schema level. It has been adopted to capture the semantic information [13, 31]. Taking the movie data in Figure 1 as an example, the relations between user and item can be revealed by the metapath User-User-Movie (UUM) for the co-user relation and User-Director-Movie (UDM) for the co-director relation. However, as Shi et al. [22] stated, metapath-based methods, heavily relying on explicit path reachability, may obtain bad performance when path connections are sparse or noisy.

¹The terms “heterogeneous information network” and “heterogeneous graph” are used interchangeably in the related literature [22, 31]. In this paper, we mainly use “heterogeneous information network” (HIN).

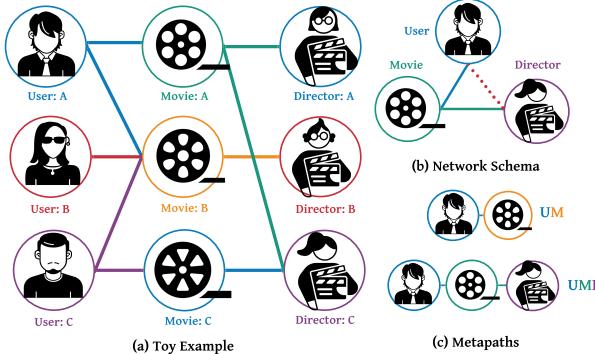


Figure 1: Heterogeneous information network and network schema. (a) movie m_A and its metapath-guided neighbors (e.g., $\mathcal{N}_{\text{UMD}}^2(u_A) = \{(u_A, m_A, d_A), (u_A, m_A, d_C), (u_A, m_B, d_B)\}$) (b) three types of nodes (user, movie, director) and two types of relations/interactions (user-movie, movie-director) in solid line, potential interactions (user-director) in dashed line. (c) metapaths involved (e.g., User-Movie (UM) and User-Movie-Director (UMD)).

Also, rich structural information of nodes outside metapaths, *i.e.*, their neighborhoods, is omitted in these approaches.

Based on the above analysis, when designing HIN, state-of-the-art methods have not well solved, even may not be aware of, the following challenges faced by HIN-based recommendation, which we address in this paper:

- (C1) How to tackle the early summarization issue? Due to the complex structures and large scales of the graphs, it is hard to make predictions directly. Hence, we consider the interactive local structures are valuable and helpful. For example, a system is recommending a user to a user u_A based on an HIN as Figure 1 shows. When we consider a candidate user such as u_B , it is natural to consider their co-movies and co-directors; namely there may exist a relationship between u_A and u_B since they share the same movie m_B . Actually, this is an example of “AND” operation between users’ neighborhoods. Also, it is easy to extend this into co-rating. In other words, the co-ratings between u_A to u_B ’s neighborhood and u_B to u_A ’s neighborhood indicate the similarity of users’ preferences, which could be helpful for recommendation. We argue that this interactive local structures are hidden and not fully utilized in previous methods.
- (C2) How to design an end-to-end framework to capture and aggregate the interactive patterns between neighborhoods? A recent attempt [13] focused on measuring the semantic proximity by interactive-paths connecting source and target nodes. However, it overlooked rich information hidden in these node neighborhoods. An HIN contains diverse semantic information reflected by metapaths [27]. Also, there are usually various nodes in different types involved in one path. Different paths/nodes may contribute differently to the final performance. Hence, besides a powerful interaction module, a well-designed aggregation module to distinguish the subtle difference of these paths/nodes and select some informative ones is required.
- (C3) How to learn the whole system efficiently? Learning interactive information on HINs is always time-consuming; especially

when faced with paths in different types and lengths for metapath-based approaches [13] and large-scale high-order information for graph-based approaches [18]. A methodology to both efficiently and effectively learn the rich interactive information on HINs is always expected.

To tackle these challenges, we propose NIRec, a neighborhood-based interaction model for HIN-based recommendation. First, we extend the definition of neighborhood in homogeneous graphs into metapath-guided neighborhood in heterogeneous graphs. Next, we design a heterogeneous graph neural network architecture with two modules to aggregate feature information of sampled neighbors in previous step. The first module, namely interaction module, constructs interactive neighborhood and captures latent information between “AND” operation. The second module, namely aggregation module, mainly consists of two components: (i) node-level attention mechanism to measure the impacts of different nodes inside a metapath-guided neighborhood, and (ii) path-level attention mechanism to aggregate content embeddings of different neighborhoods. Finally, we formulate interaction in a convolutional way and learn efficiently with fast Fourier transform (FFT). To summarize, the main contributions of our work are:

- We formalize and address an important, but seldom exploited, early summarization issue on HIN.
- We present an innovative convolutional neighborhood-based interaction model for recommendation on HINs, named NIRec, which is able to capture and aggregate rich interactive patterns in both node- and path-levels.
- We propose an efficient end-to-end learning algorithm incorporated with fast Fourier transform (FFT).

We conduct extensive experiments on four public datasets. Our results demonstrate the superior performance of NIRec over state-of-the-art baselines.

2 RELATED WORK

Heterogeneous Information Network based Recommendation. As a newly emerging direction, heterogeneous information network (HIN) [23] can naturally characterize complex objects and rich relations in recommender systems. There is a surge of works on learning representation in heterogeneous networks, *e.g.* metapath2vec [2], HetGNN [35], HIN2vec [4], eoe [32]; and their applications, *e.g.* relation inference [26], classification [36], clustering [20], author identification [1]. Among them, HIN based recommendation has been increasingly attracting researchers’ attention in both academic and industry fields. For instance, Feng and Wang [3] proposed to alleviate the cold start issue with heterogeneous information network contained in social tagged system. Metapath-based methods were introduced into hybrid recommender system in [34]. Yu et al. [33] leveraged personalized recommendation framework via taking advantage of different types of entity relationships in heterogeneous informaiton network. Luo et al. [14] proposed a collaborative filtering based social recommendation containing heterogeneous relations. Shi et al. [25] introduced weighted heterogeneous information network In [24], the similarities of both users and items are evaluated uner dual regularization framework. Recently, Hu et al. [9] leveraged metapath-based context in top-N

recommendation. As stated in [22], most existing HIN based recommendation methods rely on the path-based similarity, which may not fully mine latent features of users and items. In this paper, we introduce metapath-guided neighborhood, and propose an innovative model to capture interactive patterns hidden in neighborhoods.

Graph Representation. Graph representation learning is mainly leveraged to learn latent, low dimensional representations of graph vertices, while preserving graph structure, *e.g.*, topology structure and node content. In general, graph representation algorithms can be categorized in two types. One school is unsupervised graph representation algorithm, which aims at preserving graph structure for learning node representations [5, 17, 21, 28, 30]. For instance, DeepWalk [17] utilized random walks to generate node sequences and learn node representations. Node2vec [5] further exploited a biased random walk strategy to capture more flexible contextual structures. Struc2vec [21] constructed a multilayer graph to encode structural similarities and generate structural context for nodes. LINE [28] proposed an edge-sampling algorithm improving both the effectiveness and the efficiency of the inference. SDNE [30] implied multiple layers of non-linear functions to capture highly non-linear network structure. Another school is semi-supervised model [10, 11, 29], where there exist some labeled vertices for representation learning. For example, LANE [10] incorporated label information into the attributed network embedding while preserving their correlations. GCN [11] proposed a localized graph convolutions to improve the performance in a classification task. GAT [29] used self-attention network for information propagation, which leverages a multi-head attention mechanism. GCN and GAT are popular architectures of the general graph networks and can be regarded as plug-in graph representation modules in heterogeneous graph, such as HetGNN [35], HAN [31] and HERec [22]. In this work, our propose is not only to deliver graph representation learning techniques on heterogeneous graph, but also propose a novel efficient neighborhood-based interaction methods, which should be the first time to our knowledge.

3 PRELIMINARY

In this section, we introduce the concept of the content-associated heterogeneous information network that will be used in the paper.

Definition 3.1. Neighborhood-based Interaction-enhanced

Recommendation. The HIN-based recommendation task can be represented as a tuple $\langle \mathcal{U}, \mathcal{I}, \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{U} = \{u_1, \dots, u_p\}$ denotes the set of p users; $\mathcal{I} = \{i_1, \dots, i_q\}$ means the set of q items; $a \in \mathcal{A}$ denotes the attributes associated with objects, and $r \in \mathcal{R}$ presents the interaction behaviors between different types of objects. The purpose of recommendation is to predict the interaction $r(u_s, i_t)$, *i.e.*, click-through rate or link, between two objects, namely source user u_s and target item i_t . In our setting, interactions between source and target neighbors are leveraged to enhance the performance.

To clarify the definition of the term “neighborhood”, we model our recommendation task in the setting of **Heterogeneous Information Network** (HIN). An HIN is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which consists of more than one node type or link type. In HINs, **network schema** is proposed to present the meta structure of a network, including the object types and their interaction relations.

Figure 1(a) shows a toy example of HINs and the corresponding network schema is presented in Figure 1(b). We can see that the HIN consists of multiple types of objects and rich interaction relations. The set of objects, *i.e.*, \mathcal{U} (user), \mathcal{I} (movie) and \mathcal{A} (director); interaction relations, *i.e.*, \mathcal{R} (relation), constitute \mathcal{V} and \mathcal{E} in the HIN, respectively. The interactions here can be explained as preferences between different types of nodes, *e.g.*, $r(u_A, d_C)$; and similarities between the same type of nodes, *e.g.*, $r(u_A, u_C)$. In this case, user-item rating history, user-director preference, and movie-director knowledge constitute the interaction set \mathcal{R} . When predicting interaction $r(u_A, m_C)$ between source user u_A and target movie m_C , we consider existing interactions between their neighborhood, namely $r(m_A, d_C)$, $r(m_B, u_C)$, to help the final recommendation.

In order to capture the structural and semantic relation, the metapath [27] is proposed as a relation sequence connecting two objects. As illustrated in Figure 1(c), the User-Movie-Director (UMD) metapath indicates that users favor movies, and that these movies are guided by some directors. Based on metapaths, we introduce metapath-guided neighborhoods as follows.

Definition 3.2. Metapath-guided Neighborhood. Given an object o and a metapath ρ (start from o) in an HIN, the metapath-guided neighborhood is defined as the set of all visited objects when the object o walks along the given metapath ρ . In addition, we denote the neighbors of object o after i -th steps sampling as $\mathcal{N}_\rho^i(o)$. Specifically, $\mathcal{N}_\rho^0(o)$ is o itself. For convenience, let $\mathcal{N}_\rho(o)$ denote $\mathcal{N}_\rho^{I-1}(o)$, where I means the length of metapath. It should be noted that different from similar concept proposed in [31] utilized to derive homogeneous neighbors on heterogeneous graph, metapath-guided neighborhood here preserves semantic content since it consists of heterogeneous information.

Taking Figure 1 as an instance, given the metapath “User-Movie-Director (UMD)” and a user u_A , we can get metapath-guided neighborhood as $\mathcal{N}_\rho^1(u_A) = \{(u_A, m_A), (u_A, m_B)\}$, $\mathcal{N}_\rho(u_A) = \mathcal{N}_\rho^2(u_A) = \{(u_A, m_A, d_A), (u_A, m_A, d_C), (u_B, m_B, d_B)\}$.

Many efforts have been made for HIN-based recommendations. While, most of these works focus on leveraging graph neural networks to aggregating structural message, but overlook the effect of early summarization issue and rich interaction information. Given the above preliminaries, we are ready to introduce our NIRec model as a solution.

4 METHODOLOGY

4.1 Overview

The basic idea of NIRec is to design a neighborhood-based interaction model to enhance the representations of objects. With the help of HINs built on recommender systems, NIRec leverages metapaths to guide the selection of different-step and -type neighbors and design a heterogeneous interaction module to capture the abundant interaction message, and a heterogeneous aggregation module to obtain the rich embeddings of objects. Moreover, we represent different types of metapaths with a uniform learning procedure and conduct the fast Fourier transform (FFT) as an efficient learning algorithm.

We provide the overall framework of NIRec in Figure 2. First, we utilize the multiple-object HIN containing \langle user, item, attribute,

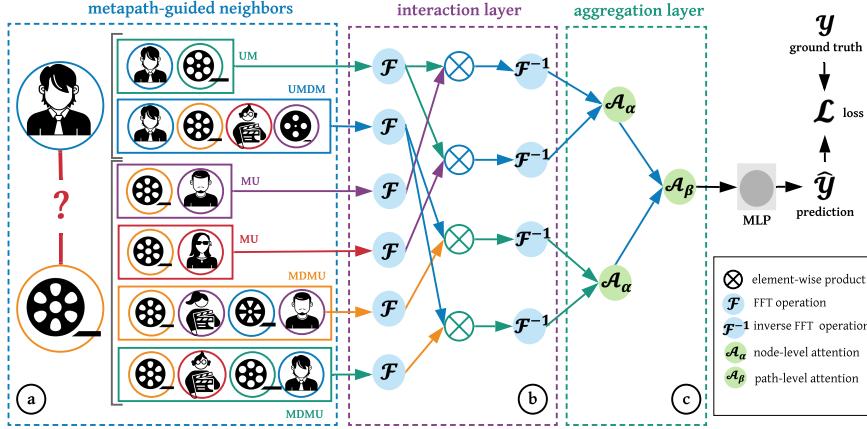


Figure 2: The overall architecture of NIRec: (a) it first samples metapath-guided neighborhood (Section 4.2); (b) next constructs interactive information via interaction layer (Section 4.3); (c) finally combines rich information through aggregation layer (Section 4.4).

relation) as the input. Second, we select metapath-guided neighbors for source and target nodes via neighbor samplings (Figure 2(a)). Third, we introduce the interactive convolutional operation to generate potential interaction information among their neighborhoods (Figure 2(b)). After that, we capture the key interactions and aggregate information via attention mechanism in both node and path level (Figure 2(c)). Finally, NIRec provides the final prediction. We illustrate the architecture in detail in the following subsections.

4.2 Neighborhood Sampling

To generate meaningful node sequences, the key technique is to design an effective random walk strategy that is able to capture the complex semantics reflected in HINs. Hence, we propose to use the metapath guided random walk method. Giving a HIN $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and a metapath $\rho : A_0, \dots, A_i, \dots, A_{I-1}$, where $A_i \in \mathcal{A}_i$ denotes i -th node guided by metapath ρ . Note that we include user set \mathcal{U} and item set \mathcal{I} in attribute set \mathcal{A} for convenience. The walk path is generated according to the following distribution:

$$P(n_{k+1} = x | n_k = v) = \begin{cases} \frac{1}{|\mathcal{N}_\rho^1(v)|}, & (v, x) \in \mathcal{E} \text{ and } v, x \in \mathcal{A}_k, \mathcal{A}_{k+1} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where n_k is the k -th node in the walk, $\mathcal{N}_\rho^1(v)$ means the first-order neighbor set for node v guided metapath ρ . A walk will follow the pattern of the metapath repetitively until it reaches the predefined length. It is worth noting that, according to *Definition 3.2*, there is no need to sample a complete metapath from source to target node.

4.3 Interaction Module

In previous HIN-based recommendations, most approaches leverage graph representation techniques to find key nodes or metapaths [31] and capture the complex structure [35]. To further mine interaction information and deal with the early summarization issue, we propose an interaction module based on metapath-guided neighbors.

Due to the heterogeneity of nodes, different types of nodes have different feature spaces. Hence, for each type of nodes (e.g., node

with type ϕ_i), we design the type-specific transformation matrix \mathbf{M}_{ϕ_i} to project the features of different types of nodes into a unified feature space. The project process can be shown as follows:

$$\mathbf{e}'_i = \mathbf{M}_{\phi_i} \cdot \mathbf{e}_i, \quad (2)$$

where \mathbf{e}_i and \mathbf{e}'_i are the original and projected features of node i , respectively. By type-specific projection operation, our model is able to handle arbitrary types of nodes.

Considering that neighbors in different distances to the source/target node usually contribute differently to the final prediction, we divide the sampled metapath-guided neighborhood into several inner-distance and outer-distance neighbor groups. As illustrated in Figure 3, when we set distance as 1, we can get inner-distance *Group 1* and outer-distance *Group 5*. In a similar way, we can obtain $2I - 1$ groups, where I denotes the metapath length. We argue that interactions should only be employed in corresponding groups. In order to perform interaction in each neighbor group, we need to face two situations. If there is only one node in *Group*, we adopt element-wise product (“AND”) operation to measure their similarity or co-ratings, e.g., $r(u_A, m_B)$ in *Group 0* case. When there is

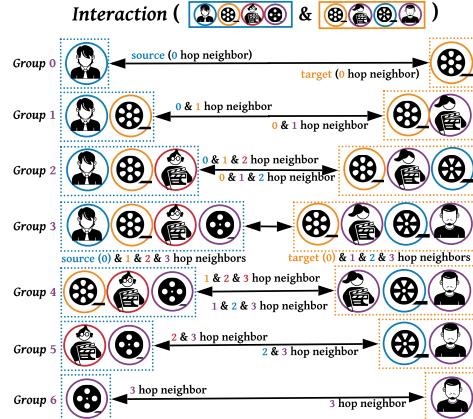


Figure 3: An illustrated example for the motivation in interaction module design. Neighborhood is grouped according to the distance to the source/target node. Interaction is only employed between corresponding neighborhoods.

more than one node, we first do interaction by production and then aggregate by summarization, e.g., $r(u_A, d_C) + r(m_B, m_B)$ in *Group 1* case.

Inspired by signal processing [16], this neighborhood division and inner group interaction can be formulated as a unified operation named convolution. Roughly speaking, the convolution mainly contains three kinds of operations, namely shift, product, and sum, which are employed repeatedly until they meet the end of the path. Take Figure 4 for example. Our task is to calculate the interaction between source user neighborhood u_A, m_B, d_B, m_D and target movie neighborhood m_B, d_C, m_C, u_C . First, we inverse the order of target movie neighborhood and obtain u_C, m_C, d_C, m_B . We shift it from left to right and observe the overlapping nodes during the shift. As shown in Figure 4(a), the first overlapping happens between source and target nodes, namely 0-hop neighbor. We utilize the product operation and obtain the co-ratings between different types of nodes $r(u_A, m_B)$ (as Figure 4(b) shows). Then, we repeatedly shift, product, and sum, and then reach the situation where all nodes are overlapped. The result in this situation is the similarity between the same type of nodes $r(u_A, u_C) + r(m_B, m_C) + r(d_B, d_C) + r(m_D, m_B)$ (as shown in Figure 4(c)). In a similar way, the last interaction happens between different types of nodes $r(m_D, u_C)$ (as shown in Figure 4(d)).

Let $H[N_\rho(o)]$ denote the embedding matrix of metapath ρ guided neighbors of object o , which can be formulated as

$$H[N_\rho(o)]_l = [e_0^\rho \oplus e_1^\rho \oplus \dots \oplus e_{I-1}^\rho], \quad (3)$$

where l represents the l -th metapath, e_i^ρ means the embedding of the node in the i -th step of one metapath, \oplus denotes the stack operation, I means the metapath length. Hence, as illustrated in Figure 5, $H[N_\rho(o)]$ is a $\mathbb{R}^{L \times I \times E}$ matrix, where L is the number of metapaths, I is the length of the metapath, E means the dimension of the node embedding. Based on convolutional operations, we further define the interaction between neighborhoods of source and target objects as

$$H[N_\rho(s), N_\rho(t)]_l = H[N_\rho(s)]_l \odot H[N_\rho(t)]_l, \quad (4)$$

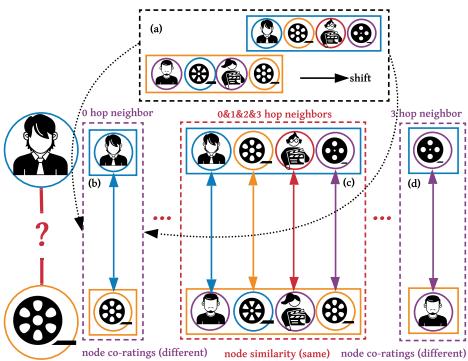


Figure 4: An illustrated example for the result after convolutional interaction operation. The result contains information of both node similarity (c) and node co-ratings (b) & (d).

where \odot denotes the convolutional operation. According to the definition of convolution, one can write formulation as

$$H[N_\rho(s), N_\rho(t)]_{l,n} = \sum_{\substack{a,b \\ a+b \bmod N=n}} H[N_\rho(s)]_{l,a} \cdot H[N_\rho(t)]_{l,b}. \quad (5)$$

One can find that $H[N_\rho(s), N_\rho(t)] \in \mathbb{R}^{L \times N \times E}$, N is the length of convolution outputs and it equals to $I_s + I_t - 1$ where I_s, I_t denote the metapath length of source and that of target nodes respectively.

The well-known convolution theorem states that convolution operations in the spatial domain are equivalent to pointwise products in the Fourier domain. Let \mathcal{F} denote the fast Fourier transformation (FFT) and \mathcal{F}^{-1} its inverse, we can compute convolutions as

$$H[N_\rho(s), N_\rho(t)] = \mathcal{F}^{-1}(\mathcal{F}(H[N_\rho(s)]) \cdot \mathcal{F}(H[N_\rho(t)])). \quad (6)$$

Let $H[N_\rho]$ denote $H[N_\rho(s), N_\rho(t)]$ in the following sections for convenience. As stated in [15], the time complexity of plain convolution is $O(I^2)$, and it is reduced to $O(I \log(I))$ when using FFT. According to the analysis above, we can conclude that not only can this structure capture both node similarity and co-ratings in grouped neighborhood, but also it can implement with high efficiency.

4.4 Aggregation Module

In this section, we consider the aggregation module in two sides. On the first side, from Figure 5, we can see that elements in interaction matrix $H[N_\rho] = H[N_\rho(s), N_\rho(t)]$, contain interactions between various types of nodes. Hence, it is natural to capture the key interaction in element/node-level during aggregation procedure. On the other side, every object o on HIN contains multiple types of semantic information represented with different metapath $\rho_0, \rho_1, \dots, \rho_{P-1}$. This further causes various interaction matrix $H[N_{\rho_0}], H[N_{\rho_1}], \dots, H[N_{\rho_{P-1}}]$. To capture key message in a complex graph, we need to fuse multiple semantics revealed by different metapath, i.e., in path/matrix-level.

Node/Element-level Attention. Similar as [31], we leverage a self-attention mechanism to learn the weights among various kinds of nodes in metapath ρ as

$$h_{ij}^\rho = (h_i^\rho W_T) \cdot (h_j^\rho W_S), \quad (7)$$

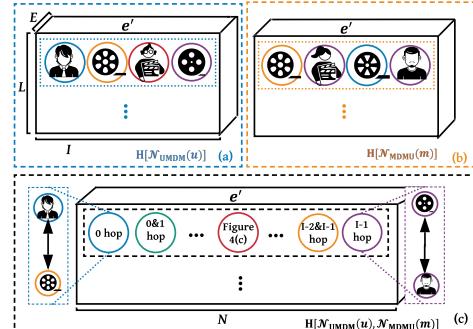


Figure 5: An illustrated example of the embedding matrix of metapath ρ guided neighborhood of source node s ($H[N_\rho(s)]$), and target node t ($H[N_\rho(t)]$) generated according to Eq. 3; and interaction matrix ($H[N_\rho(s), N_\rho(t)]$) calculated according to Eq. 5.

where h_i^ρ, h_j^ρ are elements of interaction matrix $\mathbf{H}[\mathcal{N}_\rho]$ and W_T, W_S are trainable weights. h_{ij}^ρ can show how important interaction h_j will be for interaction h_i . To retrieve a general attention value in metapath ρ , we further normalize this value in neighborhood scope as

$$\alpha_{ij}^\rho = \text{softmax}(h_{ij}^\rho) = \frac{\exp(h_{ij}^\rho / \tau)}{\sum_{j \in \mathcal{N}_\rho} \exp(h_{ij}^\rho / \tau)}, \quad (8)$$

where τ denotes the temperature factor, and \mathcal{N}_ρ here covers all the interaction elements/nodes guided by ρ , namely the entirely $\mathbf{H}[\mathcal{N}_\rho]$. To jointly attend to the neighborhoods from different representation subspaces and learn stably, we leverage the multi-head attention as in previous works [31] to extend the observation as

$$z_i^\rho = \sigma(W_q \cdot (\frac{1}{H} \sum_{n=1}^H \sum_{j \in \mathcal{N}_\rho} \alpha_{ij}^\rho (h_j^\rho W_C^n)) + b_q), \quad (9)$$

where H is the number of attention heads, and W_q, W_C, b_q are trainable parameters. Hence, the metapath based embedding z_i^ρ is aggregated based on metapath-guided neighborhood with single metapath and semantic-specific information, *i.e.*, $\mathbf{H}[\mathcal{N}_\rho]$. Given the metapath set $\{\rho_0, \rho_1, \dots, \rho_{P-1}\}$, after feeding into node-level attention, we can obtain P groups of semantic-specific interaction embeddings, denotes as $\{Z[\mathcal{N}_{\rho_0}], Z[\mathcal{N}_{\rho_1}], \dots, Z[\mathcal{N}_{\rho_{P-1}}]\}$.

Path/Matrix-level Attention. To obtain the importance of each metapath, we first transform the semantic-specific embedding through a nonlinear transformation. We then measure the path-level attention value as the average of the importance of all the semantic-specific node-level embeddings. The importance of each metapath ρ_j , denoted as ω^{ρ_j} , is shown as follows:

$$\omega^{\rho_j} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} w^T \cdot \tanh(W_q \cdot z_i^{\rho_j} + b_q), \quad (10)$$

where w is a path-level attention vector. We then normalize the above importance of all metapaths via a softmax function

$$\beta^{\rho_j} = \text{softmax}(\omega^{\rho_j}) = \frac{\exp(\omega^{\rho_j} / \tau)}{\sum_{j=0}^{P-1} \exp(\omega^{\rho_j} / \tau)}, \quad (11)$$

where τ is the temperature factor. It can be explained as the contribution of the metapath ρ_j in a specific task. With the learned weights as coefficients, we can fuse these semantic-specific embeddings to obtain the final embedding Z via

$$Z = \sum_{j=0}^{P-1} \beta^{\rho_j} \cdot Z[\mathcal{N}_{\rho_j}]. \quad (12)$$

Hence, we have obtained the final aggregation result, which involves interaction information in both node- and path-level.

4.5 Optimization Objective

The final prediction result \hat{Y} can be derived from final embedding Z via a nonlinear projection (*e.g.*, MLP). The loss function of our model is a log loss:

$$\mathcal{L}(Y, \hat{Y}) = \sum_{i, j \in \mathcal{Y}^+ \cup \mathcal{Y}^-} (y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log(1 - \hat{y}_{ij})) \quad (13)$$

where y_{ij} is the label of the instance and \mathcal{Y}^+ , \mathcal{Y}^- denote the positive instances set and the negative instances set, respectively.

4.6 Model Analysis

The learning algorithm of NIRec is given in Section A.1 in the appendix. Here we give the analysis of the proposed NIRec as follows:

Complexity. The proposed NIRec is highly efficient and can be easily parallelized. We provide the model efficiency analysis for both interaction module and aggregation module. As stated in [15], in interaction module, the complexity of the FFT-based method requires $O(L_\rho^2 I_\rho \log(I_\rho))$ where L_ρ and I_ρ denote the number of path guided by ρ and metapath length respectively. As for aggregation module, given a metapath ρ , the time complexity of node-level attention is $O(V_\rho K + E_\rho K)$, where V_ρ is the number of nodes, E_ρ is the number of metapath based node pairs, K is the number of attention heads. The computation of attention can compute individually across all nodes and metapaths. The overall complexity is linear to the number of nodes and metapath based node pairs. The proposed model can be easily parallelized, because the node- and path-level attention can be parallelized across node pairs and metapaths, respectively. The overall complexity is linear to the number of nodes and metapath based node pairs.

Interpretability. The proposed model has potentially good interpretability for the learned interaction embedding through similarity and co-ratings among neighbor nodes. Also, with the learned importance in node- and path-level, the proposed model can pay more attention to some meaningful interactions or metapaths for the specific task and give a more comprehensive description of a heterogeneous graph. Based on the attention values, we can check which interactions or metapaths make the higher (or lower) contributions for our task, which is beneficial to analyze and explain our results.

5 EXPERIMENTS

In this section, we present the details of the experiment setups and the corresponding results. To illustrate the effectiveness of our proposed model, we compare it with some strong baselines on recommendation task. We start with three research questions (RQ) to lead the experiments and the following discussions.

- (RQ1) Compared with the baseline models, does NIRec achieve state-of-the-art performance in recommendation tasks on HINs?
- (RQ2) What is the influence of different components in NIRec? Are the proposed interaction and aggregation modules necessary for improving performance?
- (RQ3) What patterns does the proposed model capture for the final recommendation decision?
- (RQ4) How do various hyper-parameters, *i.e.*, the length and type of metapath-guided neighborhood, impact the model performance?

5.1 Dataset Description

We adopt four widely used datasets from different domains, namely MovieLens movie dataset², LastFM music dataset³, AMiner paper

²<https://grouplens.org/datasets/movielens/>

³<https://grouplens.org/datasets/hetrec-2011/>

Table 1: The results of CTR prediction in terms of AUC, ACC. Note: “” indicates the statistically significant improvements over the best baseline, with p -value smaller than 10^{-6} in two-sided t -test.**

Model	Movielens		LastFM		AMiner		Amazon	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
NeuMF [7]	0.7890	0.7378	0.8900	0.8102	0.8130	0.7897	0.6841	0.6405
HAN [31]	0.8110	0.7530	0.9113	0.8289	0.8451	0.8284	0.7207	0.6831
HetGNN [35]	0.7830	0.7411	0.9020	0.8270	0.8202	0.7939	0.7061	0.6627
LGRec [8]	0.8030	0.7504	0.9127	0.8331	0.8308	0.8130	0.7058	0.6572
MCRec [9]	0.8161	0.7622	0.9274	0.8471	0.8512	0.8339	0.7274	0.6940
IPE [13]	0.8186	0.7693	0.9235	0.8440	0.8411	0.8209	0.7173	0.6789
NIRec _{CNN}	0.8342	0.7777	0.9353	0.8593	0.8734	0.8504	0.7397	0.7060
NIRec _{GCN}	0.8290	0.7630	0.9290	0.8571	0.8636	0.8475	0.7379	0.7042
NIRec	0.8468*	0.7896*	0.9404*	0.8665*	0.8760*	0.8562*	0.7493*	0.7110*

dataset⁴ and Amazon e-commerce dataset⁵. We treat a rating as an interaction record, indicating whether a user has rated an item. Also, we provide the main statistics of four datasets are summarized in Table 2 in appendix to help with reproducibility. The first row of each dataset corresponds to the numbers of users, items and interactions, while the other rows correspond to the statistics of other relations. We also report the selected metapaths for each dataset in the last column of the table. The detailed data preprocessing of is given at Section 5.1 in appendix.

5.2 Compared Methods

We use six baseline methods containing heterogeneous and attributed graph embedding models such as HAN, HetGNN, and IPE, as well as recommendation models such as NeuMF, LGRec, and MCRec. It is worth noting that, HAN, HetGNN, IPE, and MCRec are recently proposed, state-of-the-art models.

- **NeuMF**: He et al. [7] introduced a generalized model consisting of a matrix factorization (MF) component and an MLP component.
- **HAN**: Wang et al. [31] introduced hierarchical attention to capture node-level and semantic-level information.
- **HetGNN**: Zhang et al. [35] introduced an unified framework to jointly consider heterogeneous structural information as well as heterogeneous contents information, adaptive to various HIN tasks.
- **LGRec**: Hu et al. [8] proposed a unified model to explore and fuse local and global information for recommendation.
- **MCRec**: Hu et al. [9] leveraged rich metapath-based context to enhance the recommendation performance on HINs.
- **IPE**: Liu et al. [13] proposed interactive paths embedding to capture rich interaction information among metapaths.

In order to investigate the impact of different components in our model, we set several variants of NIRec model as baselines.

- **NIRec_{CNN}**: a variant of NIRec model which employs Convolutional Neural Networks (CNN) to capture contextual information within the source/target neighborhood without any interaction.
- **NIRec_{GCN}**: is another variant of NIRec model which employs Graph Convolutional Networks (GCN) to aggregate interaction information instead of attention mechanism adopted in this paper.

One can observe that NIRec_{CNN}, NIRec_{GCN} are designed to test performance gains from interaction and aggregation module, respectively. We also provide detailed configuration in Section A.3 in the appendix.

5.3 Result Analysis

We evaluate these models on the click-through rate prediction (CTR) task. We use the metrics Area Under Curve (AUC) and Accuracy (ACC), which are widely used in binary classification problems. The details of the implementation is given in Section A.4.

Experimental Results and Analysis (RQ1). The comparison results of our proposed model and baselines on four datasets are reported in Table 1. The major findings from experimental results are summarized as follows:

- Our model NIRec is consistently better than all baselines on the four datasets. The results indicate the effectiveness of NIRec on the task of CTR prediction, which has adopted a principled way to leverage interaction information for improving recommendation performance.
- Among the two kinds of baselines, most metapath-based methods (HAN, MCRec, IPE) outperform graph-based or feature-based methods (HetGNN, NeuMF) in most cases. An intuitive explanation is that those metapath-based methods can better capture the rich high-order structural information on HINs. It should be noted that our model NIRec based on metapath-guided neighborhood is able to jointly consider first-order neighbor information with high-order semantic message.
- Among HIN-based baselines, the recently proposed methods MCRec and IPE gain better performance than the others. It is easy to notice that both of them try to capture context information or interactive patterns among paths. A possible reason is that simple aggregation of semantic message on paths may lose some key information. It should be noted that NIRec not only captures interaction information but also has potentially good interpretability.

Ablation Study (RQ2). In order to investigate the contribution of each component to the final recommendation performance, we design two variants of NIRec, namely NIRec_{CNN} and NIRec_{GCN}, to study interaction and aggregation modules, respectively. The results are shown in Table 1. The findings are in two aspects. First, it (NIRec > NIRec_{CNN}) indicates that our convolutional interaction strategy is able to better capture interaction information (*i.e.*,

⁴<https://AMiner.org/data>

⁵<http://jmcauley.ucsd.edu/data/amazon/>

similarities between the same type of nodes and ratings between different types of nodes) than simply employing CNN layers. Also, it ($\text{NIRec} > \text{NIRec}_{\text{GCN}}$) shows that the attention mechanism can better utilize the metapath-based interactive information. The node-level interactions and path-level metapaths may contribute differently to the final performance. Ignoring such influence may not be able to achieve optimal performance.

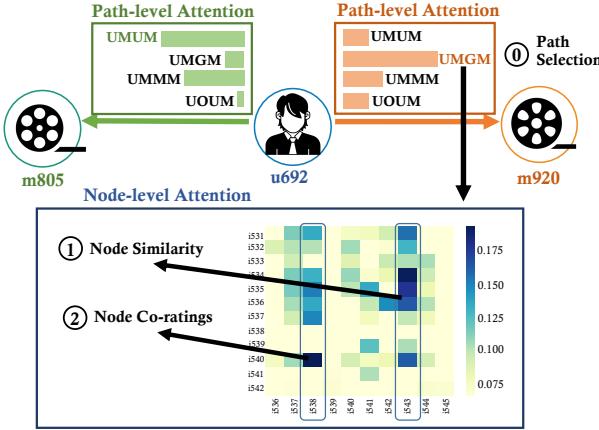


Figure 6: An illustrative example of the interpretability of interaction-specific attention distributions for NIRec. The number denotes the logic flow of interpretation.

Case Study (RQ3). A major contribution of NIRec is the incorporation of the node- and path-level attention mechanism, which takes the interaction relation into consideration in learning effective representations for recommendation. Besides the performance effectiveness, another benefit of the attention mechanism is that it makes the recommendation result highly interpretable. To see this, we select the user u692 in the MovieLens dataset as an illustrative example. Two interaction records of this user have been used here, namely m805 and m920. In Figure 6, we can see that each user-movie pair corresponds to a unique attention distribution, summarizing the contributions of the metapaths. The relation between u692-m805 pair mainly relies on the metapaths UMUM and UMMMM, while u692-m920 pair mainly relies on metapath UMGGM (denoted by ① in Figure 6). By inspecting into the dataset, it is found that at least five first-order neighbors of u692 have watched m805, which explains why user-oriented metapaths UMUM and UUUM play the key role in the first pair. As for the second pair, we find that the genre of m805 is g3, which is the favorite movie genre of u692. This explains why genre-oriented metapath UMGGM plays the key role in the second interaction. Our path-level attention is able to produce path-specific attention distributions.

If we wonder more specific reasons, we can have a look at the node-level attention. Here, we plot the interactive attention values in Figure 6. We can observe that the attention value of the similarity for the same type nodes is very high (denoted by ① in Figure 6). By inspecting into the dataset, we can find that there are metapaths connected between u692 and m920. In other words, some parts of the metapath-guided neighborhood of u692 and m920 overlap each other, which causes high similarity. Also, the co-ratings between two neighborhoods play another key role (denoted by ② in Figure 6). It is natural, since in these neighborhoods, many users are

fans of g3 and movies are in type of g3, which causes the co-ratings among them becoming really high. According to the analysis above, we can see that the distributions of attention weights are indeed very skew, indicating some interactions and metapaths are more important to consider than the others.

Impact of Metapath (RQ4). In this section, we investigate the impact of different metapaths on the recommendation performance through gradually incorporating metapaths into the proposed model. For ease of analysis, we include the NeuMF as the reference baseline. In Figure 7, we can observe that the performance of NIRec overall improves with the incorporation of more metapaths. Meanwhile, metapaths seem to have different effects on the recommendation performance. Particularly, we can find that, when adding UMGGM, NIRec has a significant performance boost in the MovieLens dataset (Figure 7(a)). Similar situation happens when adding UIVI in the Amazon dataset (Figure 7(b)). These findings indicate that different metapaths contribute differently in the final result, consistent with previous observations in Section 5.3,

Impact of N-Hop Neighborhood (RQ4). In this section, we investigate the impact of different lengths of neighborhoods, *i.e.*, n-hop neighborhood. When changing the length of the neighborhood, we make other factors, *e.g.*, metapath type fixed. For example, when the metapath is UMGGM, we study UM for length 2, UMGGM for length 4, and UMGGMGM for length 6. We conduct similar procedures for the other metapaths and obtain the results in Figure 8. We can observe that the performance first improves and then declines when the length of the neighborhood increases. Particularly, we can find that NIRec reaches the best performance in the metapath with length 4. The possible reason may be that as the length of the neighborhood increases, the metapath-guided neighborhood can maintain more information. When the length of the neighborhood is smaller than 4, the information is mainly useful for the final performance. However, when the length exceeds 4, the information includes noisy message which harms the recommendation performance. These findings indicate that different lengths of metapaths contribute differently to the final performance. Also, it should be noted that our model is able to interact and aggregate neighbors in different lengths, as illustrated in Figure 2.

6 CONCLUSION AND FUTURE WORK

In this paper, we introduced the problem of “early summarization” and proposed a neighborhood-based interaction-enhanced recommendation model, *i.e.*, NIRec, to address this problem. We first introduce the definition of metapath-guided neighborhoods to preserve the heterogeneity on HINs. Then, we elaborately designed an interaction module to capture the similarity of each source and target node pair through their neighborhoods. To fuse the rich semantic information, we proposed the node- and path-level attention mechanism to capture the key interaction and metapath, respectively. Extensive experimental results have demonstrated the superiority of our model in both recommendation effectiveness and interpretability. Currently, our approach is able to capture interactive information only in the structural (graph) side effectively. However, there is rich semantic information on both the structural (graph) and non-structural (node) sides. In the future, a promising

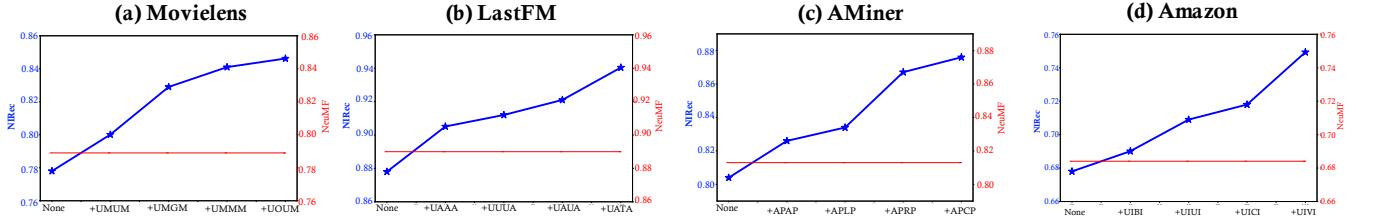


Figure 7: Performance change of NIRec when gradually incorporating metapaths in terms of AUC.

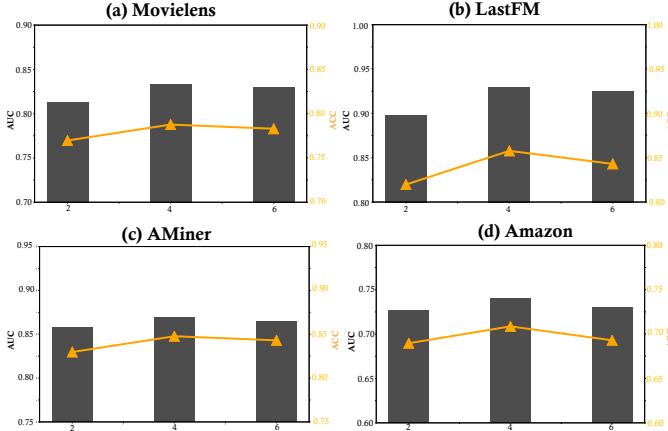


Figure 8: Performance change of NIRec with different neighborhood length in terms of AUC and ACC.

direction is extending neighborhood interaction and aggregation modules to capture key message from two sides and adapt to more general scenarios.

Acknowledgments. The corresponding author Weinan Zhang thanks the support of National Natural Science Foundation of China (Grant No. 61702327, 61772333, 61632017). We would also like to thank Wu Wen Jun Honorary Doctoral Scholarship from AI Institute, Shanghai Jiao Tong University.

REFERENCES

- [1] Ting Chen and Yizhou Sun. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In *WSDM*.
- [2] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.
- [3] Wei Feng and Jianyong Wang. 2012. Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In *KDD*.
- [4] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*.
- [5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.
- [6] Huirong Guo, Ruiming Tang, Yunning Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *IJCAI* (2017).
- [7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
- [8] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Tianchi Yang. 2018. Local and global information fusion for top-n recommendation in heterogeneous information network. In *CIKM*.
- [9] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *KDD*.
- [10] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label informed attributed network embedding. In *WSDM*.
- [11] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *ICLR* (2016).
- [12] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [13] Zemin Liu, Vincent W Zheng, Zhou Zhao, Zhao Li, Hongxia Yang, Minghui Wu, and Jing Ying. 2018. Interactive paths embedding for semantic proximity search on heterogeneous graphs. In *KDD*.
- [14] Chen Luo, Wei Pang, Zhi Wang, and Chenghua Lin. 2014. Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations. In *ICDM*.
- [15] Michael Mathieu, Mikael Henaff, and Yann LeCun. 2013. Fast training of convolutional networks through ffts. *ICLR* (2013).
- [16] Alan V Oppenheim. 1999. *Discrete-time signal processing*. Pearson Education India.
- [17] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.
- [18] Yanru Qu, Ting Bai, Weinan Zhang, Jianyun Nie, and Jian Tang. 2019. An End-to-End Neighborhood-based Interaction Model for Knowledge-enhanced Recommendation. *KDD Workshop*.
- [19] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huirong Guo, Yong Yu, and Xiuqiang He. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *TOIS* (2018).
- [20] Xiang Ren, Jialu Liu, Xiao Yu, Urvashi Khandelwal, Quanquan Gu, Lidan Wang, and Jiawei Han. 2014. Cluscite: Effective citation recommendation by information network-based clustering. In *KDD*.
- [21] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *KDD*.
- [22] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. 2018. Heterogeneous information network embedding for recommendation. *TKDE* (2018).
- [23] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *TKDE* (2016).
- [24] Chuan Shi, Jian Liu, Fuzhen Zhuang, S Yu Philip, and Bin Wu. 2016. Integrating heterogeneous information via flexible regularization framework for recommendation. *KAIS* (2016).
- [25] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. 2015. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *CIKM*.
- [26] Yizhou Sun, Jiawei Han, Charu C Aggarwal, and Nitesh V Chawla. 2012. When will it happen?: relationship prediction in heterogeneous information networks. In *WSDM*.
- [27] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB* (2011).
- [28] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*.
- [29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *ICLR* (2017).
- [30] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *KDD*.
- [31] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous Graph Attention Network. In *WWW*.
- [32] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. 2017. Embedding of embedding (eoe): Joint embedding for coupled heterogeneous networks. In *WSDM*.
- [33] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*.
- [34] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. 2013. Recommendation in heterogeneous information networks with implicit user feedback. In *RecSys*.
- [35] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *KDD*.
- [36] Yizhou Zhang, Yun Xiong, Xiangnan Kong, Shanshan Li, Jinhong Mi, and Yangyong Zhu. 2018. Deep collective classification in heterogeneous information networks. In *WWW*.

A APPENDIX

A.1 Pseudocode of NIRec Training Procedure

The pseudocode of the NIRec training procedure is described in Algorithm 1. In the HIN-based recommendation scenario, each source node n_s corresponds to a user u_s , and each target node n_t corresponds to an item i_t . The task in this work can be either stated as link predictions on HINs or CTR predictions on HIN-based recommendation.

Algorithm 1 NIRec

INPUT: HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; node feature $\{e, i \in \mathcal{V}\}$; metapath set $\{\rho_0, \rho_1, \dots, \rho_{P-1}\}$; source node n_s and target node n_t

OUTPUT: final link prediction \hat{Y} between n_s and n_t

```

1: Initialize all parameters.
2: repeat
3:   for each metapath  $\rho_k \in \{\rho_0, \rho_1, \dots, \rho_{P-1}\}$  do
4:     Find metapath-guided neighborhoods of  $n_s, n_t: \mathcal{N}_{\rho_k}(n_s), \mathcal{N}_{\rho_k}(n_t)$  according to Eq. 1.
5:     Obtain interaction result  $H[\mathcal{N}_{\rho_k}(n_s), \mathcal{N}_{\rho_k}(n_t)]$  according to Eq 6.
6:     Calculate node/element-level embedding  $z^{\rho_k}$  according to Eq. 9.
7:   end for
8:   Fuse path/matrix-level embedding  $Z$  according to Eq. 12.
9:   Obtain final predication  $\hat{Y}$  via MLP.
10:  Calculate loss  $\mathcal{L}(Y, \hat{Y})$  according to Eq. 13, and Back propagation.
11: until convergence

```

As Algorithm 1 shows, we first generate neighborhoods $\mathcal{N}_{\rho_k}(n_s), \mathcal{N}_{\rho_k}(n_t)$ of n_s, n_t guided by metapath ρ_k in line 4. We then obtain interactive information $H[\mathcal{N}_{\rho_k}(n_s), \mathcal{N}_{\rho_k}(n_t)]$ through the interaction module described in Section 4.3 in line 5. To fuse the interaction results, we leverage the node-level attention mechanism to obtain representation z^{ρ_k} in line 6. We repeat the above procedure for different metapaths and obtain various node-level embeddings $z^{\rho_0}, z^{\rho_1}, \dots, z^{\rho_{P-1}}$. To capture the key metapath in the current task, we employ the path-level attention mechanism to get embedding Z in line 8. Finally, we obtain the final predication \hat{Y} via MLP layers in line 9.

A.2 Data Preprocess

Each dataset is processed as follows. Since the relations between users and items are originally in rating format, we convert ratings to binary feedbacks: ratings with 4–5 stars are converted to positive feedbacks (denoted as “1”), and other ratings are converted to negative feedbacks (denoted as “0”). After the datasets are processed, we split each dataset into training/validation/test sets at a ratio of 6:2:2.

A.3 Model Variants Configuration

- **NIRec_{CNN}**: This variant does not consider the interpretability in the interaction module. That is, it directly fuses the neighborhood of each source and target node via Convolutional Neural

Networks (CNN). This context embedding method is quite similar with the path instance embedding technique in [9]. The aggregation module is the same as NIRec. Hence, this variant is designed to show the performance gain by our interaction module.

- **NIRec_{GCN}**: This variant replaces the attention mechanism of NIRec with a standard Graph Convolutional Network (GCN). That is, both the node- and path-level features are fed to GCN layers to get content embeddings without fully considering different types of nodes and metapaths on HINs. The interaction module is the same as NIRec. Hence, this variant is designed to show the performance gain by our aggregation module.

Table 2: Statistics of the four datasets. The last column reports the selected metapaths in each dataset.

Datasets	Relations (A-B)	A	B	A-B	Metapath
Movielens	User-Movie	943	1,682	100,000	UMUM
	Movie-Movie	1,682	1,682	82,798	UMMM
	User-Occupation	943	21	943	UOUM
	Movie-Genre	1,682	18	2,861	UMGM
LastFM	User-Artist	1,892	17,632	92,834	UAUA
	User-User	1,892	1,892	18,802	UUUA
	Artist-Artist	17,632	17,632	153,399	UAAA
	Artist-Tag	17,632	11,945	184,941	UATA
AMiner	Author-Paper	164,472	127,623	355,072	APAP
	Paper-Conference	127,623	101	127,632	APCP
	Paper-Label	127,623	10	127,623	APLP
	Paper-Reference	127,623	147,251	392,519	APRP
Amazon	User-Item	3,584	2,753	50,903	UIUI
	Item-View	2,753	3857	5,694	UIVI
	Item-Brand	2,753	334	2,753	UIBI
	Item-Category	2,753	22	5,508	UICI

A.4 Implementation Details

Hyper-parameters. The embedding dimension of NIRec is set to 128. As stated in Section 4.2, we perform the metapath-guided random walk method. The size of sampled neighbors set of each node equals to 20 guided by the current metapath. The length of metapath-guided neighborhood equals to the length of the metapath. We investigate the impacts of different lengths of neighborhoods, *i.e.*, n-hop neighborhoods in Section 5.3.

Baseline settings. For fair comparisons, we employ the proposed method and baselines as following: (i) In the data processing procedure, we sample and store paths guided by metapaths in a unified framework. For instance, when sampling data from Movielens following UMUM-path, we can obtain 10 metapaths between u_0 and m_0 , *i.e.*, $\rho(u_0 - m_0)$, as well as 10 metapath-guided neighborhoods of u_0 , *i.e.*, $\mathcal{N}(u_0)$, and 10 metapath-guided neighborhoods of m_0 , *i.e.*, $\mathcal{N}(m_0)$. That is, the first node of both metapath $\rho(u_0 - m_0)$ and metapath-guided neighborhood $\mathcal{N}(u_0)$ is the same source node u_0 . The difference is that the metapath must end with target node m_0 while the neighborhood can end with any node in the type of movie. (ii) For metapath-based baselines such as MCRec and IPE, we feed models with the same paths. For GNN-based baselines, such as HAN and HetGNN, we generate n-hop neighbor nodes from sampled paths. (iii) The embedding dimensions of all baselines are set to 128 (same as NIRec).