Neurocomputing 461 (2021) 438–449

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Path-based reasoning over heterogeneous networks for recommendation via bidirectional modeling

Junwei Zhang [a,b], Min Gao [a,b,*], Junliang Yu [c], Linda Yang [d], Zongwei Wang [a,b], Qingyu Xiong [a,b]

[a] Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, Chongqing 401331, China
[b] School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China
[c] School of Information Technology and Electrical Engineering, The University of Queensland, Queensland 4072, Australia
[d] School of Engineering, University of Portsmouth, Portsmouth PO1 3AH, UK

ARTICLE INFO

Article history:
Received 7 June 2020
Revised 19 June 2021
Accepted 15 July 2021
Available online 20 July 2021
Communicated by Zidong Wang

Keywords:
Recommender systems
Heterogeneous information network
Path-based reasoning

ABSTRACT

Heterogeneous Information Network (HIN) is a natural and general representation of data in recommender systems. Combining HIN and recommender systems can not only help model user behaviors but also make the recommendation results explainable by aligning the users/items with various types of entities in the network. Over the past few years, path-based reasoning models have shown great capacity in HIN-based recommendation. The basic idea of these models is to explore HIN with predefined path schemes. Despite their effectiveness, these models are often confronted with the following limitations: (1) Most prior path-based reasoning models only consider the influence of the predecessors on the subsequent nodes when modeling the sequences, and ignore the reciprocity between the nodes in a path; (2) The weights of nodes in the same path instance are usually assumed to be constant, whereas varied weights of nodes can bring more flexibility and lead to expressive modeling; (3) User-item interactions are noisy, but they are often indiscriminately exploited. To overcome the aforementioned issues, in this paper, we propose a novel path-based reasoning approach for recommendation over HIN. Concretely, we use a bidirectional LSTM to enable the two-way modeling of paths and capture the reciprocity between nodes. Then an attention mechanism is employed to learn the dynamical influence of nodes in different contexts. Finally, the adversarial regularization terms are imposed on the loss function of the model to mitigate the effects of noise and enhance HIN-based recommendation. Extensive experiments conducted on three public datasets show that our model outperforms the state-of-the-art baselines. The case study further demonstrates the feasibility of our model on the explainable recommendation task.

1. Introduction

Networks such as social networks, biological networks, and transportation networks connect all kinds of data in our lives. The objects and interactions in the real world are often multi-modal and multi-type. To capture and utilize such heterogeneity of nodes and links, some researchers propose heterogeneous information networks (HIN) in many practical network mining scenarios, especially in recommender systems (RS). HIN-based models can not only alleviate data sparsity problem and improve the performances of models but also make the recommendation results explainable [47,16,17,24], because the connections in HIN can intuitively reflect the extra connectivity information between users and items. Besides, these connections provide algorithm designers with a new method of debugging and improve the transparency of the recommendation model and users' cohesion. Due to their significant advantages, HIN-based RS have received extensive attention from academia and industry.

To explore the potential of HIN in RS, one line of research pays attention to making recommendations using embedding models [25,1,20,40], which can be divided into two categories: node similarity and path similarity based models. The basic idea of the former is to align heterogeneous graphs in a regularized vector space and reveal the similarity between nodes by calculating the distance between representations of nodes, such as TransE [20] and node2vec [6]. Although these models have achieved some performance improvement, they lack the consideration of discovering multi-hop relational paths. Another embedding research in RS mainly integrates heterogeneous nodes and edges with predefined

* Corresponding author.
E-mail address: gaomin@cqu.edu.cn (M. Gao).

https://doi.org/10.1016/j.neucom.2021.07.038
0925-2312/

path schemes and calculates the similarity between paths to learn the low-dimensional representation vector of nodes, such as meta2vec [3] and IF-BPR [40]. Despite their prevalence and effectiveness, they can hardly cover all real-world situations because the design of the path schemes often requires the extensive domain knowledge. Briefly, the above-mentioned models based on network embedding restrict the power of node relationships in HIN.

Some researchers are aware of the adverse factors of HIN embedding and propose to conduct explicit reasoning over HIN to make recommendations instead of merely embedding the network as vectors for similarity matching. Reasoning is a process of deducing new knowledge from prior knowledge. Unlike the models based on HIN embedding, the reasoning model over HIN usually use the random walk to construct node sequences from HIN, select some nodes in the path as prior knowledge, and predict the unknown nodes drawing on sequence modeling. With the rise of machine learning, some studies have made attempts to introduce machine learning technology into the field of reasoning, such as using Markov chain [7,8,23], recurrent neural networks [11,12,29], and attention mechanisms [33,21,46,19]. Although the above methods have made some improvements in the recommendation performance, they still have a significant limitation. Concretely, most of these models use unidirectional models from left to right to learn the node sequences of HIN, resulting in the inability to simulate complex relationships between nodes. The complex relationships in HIN can be attributed to: (1) The same node may have different meanings in different paths. (2) The importance of nodes in a path is likely to be different. (3) User-item interactions are noisy because users may mis-click some items when they are browsing items. If the complex node relationship cannot be simulated, the learned node vector will be inaccurate, which will ultimately affect the recommendation effect.

In this paper, we propose an **a**ttention-based **b**idirectional **L**ong-Short Term Memory Network (LSTM) with **a**dversarial learning for recommendation model based on path reasoning over **H**IN, named **ABLAH**. Specifically, to better learn the complicated relationship between nodes in a heterogeneous network path, we learn the nodes' sequence by bidirectional node sequences modeling. We take the acquired path as input and use the bidirectional LSTM to model the path. Meanwhile, to reflect the importance of different nodes in a path, we use the attention mechanism to learn the weights of different nodes. To mitigate noise relationship between selected path nodes, we innovatively introduce adversarial learning to optimize the loss function. Adversarial learning is an emerging model training method that can effectively identify noise in interactive data by designing a discriminator network model. In our work, we add an adversarial regularization term to optimize the vectors learned through the bidirectional LSTM. Specifically, we designed a minimax function that maximizes the adversarial regularization term's effect and minimizes the loss function of the ABLH (bidirectional LSTM based on attention). This method can alleviate the noise problem and the high time complexity problem caused by repeated training of the ABLH model. Like some recent work [11,12], the type of recommendation problem we are considering in this work is Top-N recommendation. We predict the probability of users purchasing items and recommend Top-K items to users based on the calculated probability. We focus on implicit feedback because it can be tracked automatically and is much easier to collect. Compared to explicit feedback (i.e., ratings and reviews), implicit feedback can indirectly reflect users' preferences through their behaviors like watching videos, purchasing products, and clicking items. In summary, we have made the following contributions:

1. We design a novel path-based reasoning approach using attention-based bidirectional LSTM for recommendation over HIN.
2. To the best of our knowledge, we are the first to combine the adversarial regularization term and the embedding of HIN to alleviate the problem of noise relationships in recommendation.
3. We conduct extensive experiments on multiple public datasets to demonstrate the superiority of our model. The results show that it outperforms some state-of-the-art recommendation models, and the case study shows the feasibility of our model on the explainable recommendation task.

In the rest of this paper, we further discuss the related work in Section 2. In Section 3, we present our ABLAH in detail. We describe experimental research in Section 4. Finally, we summarize our work and look forward to the future work in Section 5.

2. Related work

This paper focuses on explaining the recommendation reason using HIN and mitigating the data noise problem. In this section, we discuss the work related to HIN representation learning and knowledge reasoning over the graph. Apart from these, adversarial learning is also essential to the work for mitigating the data noise. Hence, we will review the previous work from three aspects in this section: models based on heterogeneous information networks representation learning, knowledge reasoning methods over the graph, and adversarial learning methods.

2.1. Heterogeneous Information Networks Representation Learning (HIN-based RS)

Recently, some researchers realize the importance of HIN in recommendation models because HIN includes the abundant information about users and items. To make full use of relational heterogeneity, Yu et al. [42] introduced potential features based on meta-paths and used Bayesian ranking optimization to represent the connectivity between users and items in different types of paths. Then, they defined the recommendation model at the global and personalized levels. Shi et al. [26] implemented a semantic-based recommendation model, called HeteSim, to measure nodes' relevance. Wang et al. [5]propose to use multiple types of information to build a HIN, mapping different types of edges to different feature sets and learning the weight of each edge. Yuan et al. [43] use social information to explore the membership's social relationship and use two fusion strategies, regularization, and collective matrix decomposition, in HIN-based recommendation models. Besides, Sun et al. [27] introduce the concept of meta-path similarity to a HIN-based model. Yu et al. [40] identify more reliable friends with similar preferences by designing meaningful meta-paths for users to optimize the social BPR model [48]. Similarly, some studies [2,3,25] also use the random walk technique based on meta-path to construct the sequences of nodes and learn the vector representation of nodes to improve the RS's performance. However, most of these methods rely on path-based similarity, whereas the paths should be designed in advance. It is impractical to cover all possible meta-paths in large scale HIN.

In recent years, with the popularity of deep learning, some researchers have tried to use related technologies to mine the connection relationship between nodes in heterogeneous networks. For example, some researchers take HIN graph data as an input, design graph neural network architectures to generate embedding representations for each node, and consider the impact of different adjacent groups for RS [45,35,4]. Nevertheless, these models ignore the influence of noise in the network on the final effect. The techniques and limitations of the models about HIN-based RS introduced above are summarized in Table 1. Considering the aforementioned approaches' limitations, we utilize the random walk to extract reasoning paths that fix the first and last node and apply adversarial learning to migrate the adverse effect of data noise.

2.2. Knowledge reasoning over graph

Recent years, the research on graph-based reasoning has received widespread attention because reasoning is an important form of simulated thinking, which can deduce some conclusions from existing knowledge. The reasoning models include rule-based knowledge reasoning models and neural network-based ones. These models' basic idea is to apply simple rules or statistical features to reason some candidate facts. For instance, a reasoning system called Never-Ending Language Learning system (NELLs) [28] infers a new relationship instance from other learned rules. Davis et al. [18] theoretically study the suitability of learning the weights of a Markov logic network from a knowledge graph in the presence of missing data.

The neural network has a strong ability to learn the feature representation by converting the input data's feature distribution from the original space to another feature space. Therefore, it is suitable for abstract tasks such as knowledge reasoning. Tay et al. [32] first proposed a multi-task neural network (MT-KGNN) to learn the representation of nodes, relationships, and attributes in an inference process. MT-KGNN is composed of RelNet and AttrNet, where RelNet models the structure and relationships of graphs, and AttrNet models nodes and corresponding attributes. Wang et al. [38] introduce the attention mechanism into graph reasoning to solve the problem that different nodes have various importance in multi-hop reasoning. Among the related works, the most relevant study is the Knowledge Perception-based Path Recurrent Network (KPRN) [36], which learns the low-dimensional vector representation of nodes and their relationships in the graph and infers user preferences using unidirectional LSTM. Although better reasoning performance can be obtained with deep learning, these methods ignore the complicated relationship between nodes in paths, such as the importance of nodes and the same node's multiple meanings. We summarize the techniques and limitations of the models for knowledge reasoning on the graph, as shown in Table 2. Hence, this paper tries to apply attention-based bidirectional LSTM to reason recommendation results over HIN.

2.3. Adversarial learning

Adversarial learning is an emerging paradigm for solving data noise and improving model robustness. As deep learning is widely used in various fields such as recommendation systems, its reliability and stability have attracted widespread attention, especially when it suffers from simple disturbances. Besides that, there is always noise data in the interaction information that affects the recommendation effect due to accidental clicks or random browsing of certain items. Hence, the RS fails to achieve the desired result. The basic idea of adversarial learning is to design a maximum and minimum loss function. Related research can be divided into two directions: using the generative adversarial network (GAN) and adding a regularization term to the vector representation. The first model that applies GANs to RS is IRGAN [34], which attempts to unify the generative and discriminative information retrieval models under a framework. Reliable social RS based on GAN [41] was proposed to alleviate data sparsity and improve social recommenders' performance by generating more reliable friends. The experimental results verify that it outperforms all other methods in ranking prediction. However, recommendation models based on generative adversarial networks usually require designing discriminator and generator networks, and much training time.

He et al. [9] verify that adding adversarial disturbances to RS can improve the original system's robustness. They propose APR, which forces the model to consider the deviation caused by noise in advance. Tang et al. [30] extend APR into image recommendation to improve image-based recommendation models' robustness. The above work show that adversarial regularization (disturbances) can enhance the model's robustness and effectively alleviate data noise. The main technologies involved in the above models are summarized in Table 3. However, there are few studies combining the thought of adversarial learning with the knowledge reasoning. Hence, this paper tries to apply adversarial learning to reduce the impact of data noise in the reasoning.

3. The proposed model

In this section, we introduce the structure of our model. First, we formalize the notations of HIN and the HIN-based recommendation task. Then we introduce the various components of the proposed model, ABLAH, which consists of ABLH and adversarial regularization term. Finally, we discuss the model size and the time complexity of our model. Fig. 2 shows the schematic overview of ABLAH.

3.1. Preliminary

As a special kind of network, HIN includes multiple types of nodes and edges. In this subsection, we introduce the main notations involved in this paper and formalize the problem of HIN-based recommendation.

Heterogeneous Information Networks (HIN). We formally use $G = (V, E)$ to represent heterogeneous information networks (HIN), where V is the set of nodes and E is the set of edges. In the HIN, each node v and each edge e have a mapping relationship $\varphi(v) : V \rightarrow T_V, \varphi(e) : E \rightarrow T_E, T_V + T_E \geqslant 3$, where T_V and T_E are the set of types of nodes and edges, respectively.

Path in HIN. Within G, we treat the node sequences from user u to item i as a path, which is defined as $p = [v_1, v_2, \ldots, v_L]$, where L is the maximum number of nodes in a path. Different from the handcrafted meta-path, we fix the user node u and the target node i as the first and the last node in the path respectively, whereas other nodes are extracted by random walk. An example of HIN about music is illustrated in Fig. 1. There are five types of nodes in the heterogeneous music network: user, music, album, singer, and song category. For example, Tony and Tom are user, Eagles is a singer, Folk is a song category, and "Live New York" is an album. Besides that, "Right Here Waiting", "In the City", "The Sad Café",

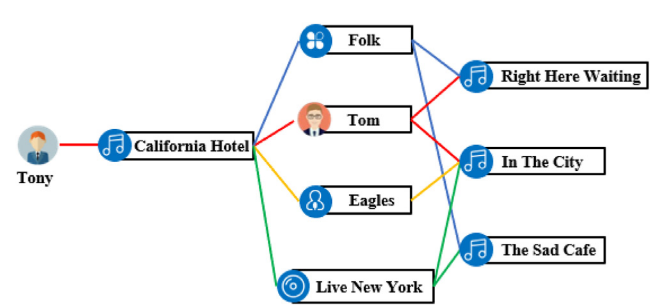

Fig. 1. Illustration of recommendation in the music domain. The lines between nodes are the corresponding relations.

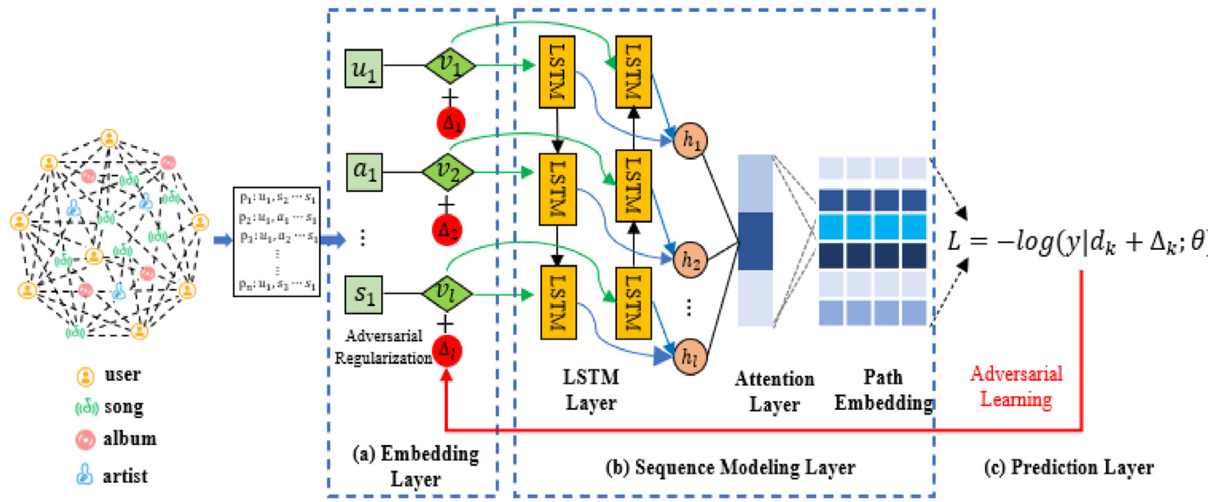

Fig. 2. The overview of our proposed framework for enhanced HIN-based recommendation. (a) Embedding layer. (b) Sequence modeling layer. (c) Prediction layer.

and "California Hotel" are songs. Connections of different colors represent various relationships between nodes. Red links represent the relationship between users and songs, such as the relationship between Tony and "California Hotel"; green links represent the affiliation between songs and albums, e.g. such as "California Hotel" belongs to the album "Live New York"; blue lines represent the affiliation between songs and categories, e.g. "California Hotel" belongs to the category Folk; the yellow lines represent the singing relationship between songs and singers, for example: "Eagles" play California "Hotel".

1. $p_1 = [Tony, CaliforniaHotel, Tom, InTheCity]$,
2. $p_2 = [Tony, CaliforniaHotel, Eagles, InTheCity]$,
3. $p_3 = [Tony, CaliforniaHotel, LiveNewYork, InTheCity]$,

HIN-Based Recommendation Task. The path set can be formalized as $P(u,i) = \{p_1, p_2, \ldots, p_k\}$, which take the target user u as the head node and the target item i as the tail node. The task is to estimate the probability that the user will buy or like the item. We obtain the probability through the paths between the user and the item, which can be computed as

$$\hat{y}_{ui} = f_\theta(u, i|P(u,i)), \tag{1}$$

where $\hat{y}_{ui}$ represents the interaction probability between the user and the item, and f denotes the mapping function with the parameter θ.

3.2. The architecture of ABLH

The main idea of this paper is to model node sequence information of HIN and capture the complex relationships between nodes in HIN. Hence, we firstly focus on the attention-based bidirectional LSTM for reasoning and introduce the basic solution (ABLH) to achieve this goal. The input of ABLH is a paths set about each user-item pair, and the output is a score of probability that the user might interact with the target item. As illustrated in the Fig. 2, our model contains three main parts: (a) **The Embedding Layer:** This module maps the value and type of the node into low-dimensional vector representations to obtain the initialization vectors for the downstream task; (b) **The Sequence Modeling Layer:** We adopt the attention-based bidirectional LSTM to capture the contextual relationship between the nodes in a path. To be specific, both the predecessor node and the subsequent nodes on the path may influence each other and the weights of nodes in the same path could be different; (c) **The Prediction Layer:** To combine the representation of nodes from two-way sequences and predict the probability of user interaction on the target item, we adopt two-layer feed-forward neural network to obtain the score.

Embedding Layer. We conduct a random walk over heterogeneous network by probabilistically selecting next node v_{i+1} from a current node v_i. Concretely, we define the transition probability as follows:

$$P_{\text{trans}}(v_{i+1}^q|v_i) = N(v_i) \cap N(v_{i+1}^q)/N(v_i), \tag{2}$$

where $N(v_i)$ represents the neighbors connected to the node v_i. v_{i+1}^q denotes the next node that belongs the type q. In this way, we can avoid repeating the same node in the path generated by the random-walk. For different datasets, the number of nodes in the path generated by the random walk is different, which means the number with the best recommendation effect is selected. The specific number is introduced in detail in the experiment in Section 4.4. In each path, the head is the user u and the tail is the item i. We map types of nodes and the specific value of nodes into two vectors $e_l^t \in \mathbb{R}$ and $e_l^s \in \mathbb{R}$. Each representation captures different meanings with respect to different perspectives. To fully exploit the nodes' type and specific value, we concatenate these two embedding to form the final node embedding h_l. For each node in the path, the new initialization vector of the node h_l can be obtained according to Eq. (3):

$$h_l = e_l^t + e_l^s. \tag{3}$$

In the HIN, different node types have different meanings. Therefore, to highlight the differences between various types of nodes, we use two vectors to initialize the type of node l and its vector value, namely e_l^t and e_l^s. The representation of each node is obtained by adding the type vector and the concrete value vector. Through the embedding layer, we can obtain an embedding $[h_1, h_2, \ldots, h_L]$ for the path p_k, where each element represents the vector of a node.

Sequence Modeling Layer. Through HIN-based embedding, we obtain the vector of nodes. For each user-item pair, we take the embedding of nodes h_l in paths as input, and the probability of user's favorite item is obtained through the attention-based bidirectional LSTM. We use the attention-based bidirectional LSTM to further infer the final target item due to its capability of capturing the complex sequence information deeply among nodes. Different from the sentences in the NLP problem, the number of nodes in path of HIN is finite. We need to pay more attention to the importance of different nodes in the path and their impact on the entire path. Compared with traditional unidirectional modeling model, our bidirectional one considers both left and right context in a path to achieve context-based inference. As shown in Fig. 2, the output of the model is jointly determined by the hidden state of the forward LSTM and backward LSTM. Specifically, in the forward LSTM, the target item i, which is the last node in the path p, is represented as h_L. The low-dimensional vector of the previous node is represented as h_{L-1}. The hidden state of the node is $\overrightarrow{h}_{L-1}$ and its cell state vector is defined as $\underset{c}{\rightarrow}_{L-1}$. We define Eq. (5) to learn the hidden state of target item $\overrightarrow{h}_L$:

$$\overrightarrow{z_L} = tanh\left(\overrightarrow{W_z}h_{L-1} + \overrightarrow{W_h}\overrightarrow{h}_{L-1} + \overrightarrow{b_z}\right) \tag{4}$$

$$\begin{aligned} \overrightarrow{f_L} &= \sigma\left(\overrightarrow{W_f}h_{L-1} + \overrightarrow{W_h}\overrightarrow{h}_{L-1} + \overrightarrow{b_f}\right) \\ \overrightarrow{i_L} &= \sigma\left(\overrightarrow{W_i}h_{L-1} + \overrightarrow{W_h}\overrightarrow{h}_{L-1} + \overrightarrow{b_i}\right) \\ \overrightarrow{o_L} &= \sigma\left(\overrightarrow{W_o}h_{L-1} + \overrightarrow{W_h}\overrightarrow{h}_{L-1} + \overrightarrow{b_o}\right) \\ \overrightarrow{c_L} &= \overrightarrow{f}_L * \overrightarrow{c_{L-1}} + \overrightarrow{i}_L * \overrightarrow{z_L} \\ \overrightarrow{h}_L &= \overrightarrow{o_L} * tanh\left(\overrightarrow{c_L}\right), \end{aligned} \tag{5}$$

where $z \in \mathbb{R}$ represents the transformation module, $\overrightarrow{i_L}, \overrightarrow{o_L}$, and $\overrightarrow{f_L}$ represent input, output, and forget gate respectively. $\overrightarrow{W_z}, \overrightarrow{W_i}, \overrightarrow{W_f}$, and $\overrightarrow{W_o}$ are mapping coefficient matrices, while $\overrightarrow{b_z}, \overrightarrow{b_i}, \overrightarrow{b_f}$, and $\overrightarrow{b_o}$ are bias vectors. $\sigma(\cdot)$ is the activation function, and $(*)$ means the elements-wise multiplication. The backward LSTM layer only needs to take the opposite node sequences as input to obtain the hidden state $\overleftarrow{h_L}$. In this way, we can make full use of the forward and backward information in the path. Finally, the representation vector of h_L is calculated by concatenating the hidden state vectors generated in two directions $\overrightarrow{h}_L$ and $\overleftarrow{h_L}$, as shown in Eq. (6):

$$\overline{h_L} = \left[\overrightarrow{h}_L \oplus \overleftarrow{h_L}\right]. \tag{6}$$

Obviously, not all nodes contribute equally to the representation of a path, some nodes are more important than others. Yuan et al. [44] have set a good example to capture the sequential information among different nodes in RS by using LSTM, but they did not consider the importance of nodes in a path. Hence, we highlight those valuable nodes via attention mechanism and the enhanced path representation is computed as shown in Eq. (6):

$$\begin{aligned} M_u &= tanh(\overline{H_u}), \\ \alpha_u &= softmax(W_u M_u), \\ R_u &= \overline{H}_u \alpha_u^T, \end{aligned} \tag{7}$$

where $\overline{H_u}$ is the hidden state representation matrix of all path nodes for the user u, α_u is the attention matrix, and W_u is the coefficient matrix. We use weighted sum of the output vectors to represent the vector of the path R_u.

We first use a two-layer fully connected network to further optimize the representations, as shown in Eq. (8):

$$s_k = W_1^T ReLu\left(W_2^T R_u\right), \tag{8}$$

where W_1^T and W_2^T are the coefficient matrix of the forward neural network respectively. Since there are multiple paths between the user u and the item i, we average the path vectors as the final representation vector of the path s_{ui}:

$$s_{ui} = \frac{1}{K}\sum_{k=1}^{K} s_k. \tag{9}$$

Prediction Layer. By stacking bidirectional sequence modeling layer, the path representation are capable of receiving the information propagated from right and left node sequences. $\hat{y}_{ui}$ is probability score of user interaction on items in each path. Given an instance $(u,i), \hat{y}_{ui}$ can be computed as follows:

$$\hat{y}_{ui} = \sigma(s_{ui}), \tag{10}$$

where σ is the activation function.

Similar to the work of He et al. [9], we consider the recommendation task as a classification problem in which the target item is the label. Therefore, we design the following loss function to learn the parameters of our model:

$$L = -\sum_{(u,i)\in\vartheta^+} \log \widehat{y_{ul}} + \sum_{(u,j)\in\vartheta^-} \log\left(1 - \widehat{y_u}\right), \tag{11}$$

where θ^+ denotes the positive interaction pairs, θ^- represents the negative samples.

3.3. Enhance framework with adversarial learning

We note that it is not straightforward to train the ABLH for reasoning. This is mainly because the above model ignores the effect of noise in the interaction data. In RS, users sometimes click on items beyond their interests, which leads to noisy edges that should not exist in HIN. To cope with the noise problem and improve the accuracy of the node representations, we propose ABLAH, which refines ABLH with adversarial learning. Concretely, inspired by the existing models based on adversarial learning [9,30], we design a new loss function and optimize it to achieve the above purpose. Typically, the adversarial regularization term is applied to either feature representation or model parameters. An intuitive choice is to apply term to model parameters. However, this method is difficult to add the regularization term to the appropriate parameter position during the training process because the model structure is complicated. Moreover, the latter solution tend to cause the model over-fitting problem because the interaction information is sparse and vast. To avoid these difficulties, we add the adversarial regularization term to the node embedding representation vector. Explicitly, we define the objective function as Eq. (12):

$$\begin{aligned} &L_{adv} = -logp(y|s_{ui} + \Delta_k; \theta), \\ &where,\ \Delta_k = -\epsilon g/\|g\|_2, \\ &g = \bigtriangledown_{s_{ui}} logp(y|s_{ui}; \theta), \end{aligned} \tag{12}$$

where Δ_k represents the adversarial regularization term, ϵcontrols the size of Δ_k, and the loss function is finally calculated by back-propagation.

Algorithm 1 ABLAH

Input: network $G = (V, E)$, node embedding dimension d, learning rate λ, the number of paths per user K, the number of nodes in each path L, coefficient ϵ, the number of neurons r, the batch size b, and the number of iterations *iter*;
Output: Ensemble of classifiers on the current batch, E_n;
1: Initialize all the model parameters θ, initialize the coefficient matrix W and bias vector b;
2: Utilize random walks to generate paths for each user, where the head node is u and the tail node is i;
3: Initialize the node vector h_l;
4: **repeat**
5: Adopt the bidirectional LSTM model to achieve path-based reasoning over HIN using Eqs. (5) and (6);
6: Adopt the attention mechanism to capture different

a (*continued*)

Algorithm 1 ABLAH

importance of nodes using Eq. (7);
7: Calculate the hidden state of the target item $\overline{h}_L$;
8: Update the coefficient matrix W and bias vector b using Eq. (11);
9: Calculate objective function L_{adv} and update the adversarial regularization term Δ_k using Eq. (12);
10: **until** ($k < K$ and $l < L$ and $u < m$)

3.4. Discussion

Our model integrates multiple components to cope with different issues. We summarize our algorithm in Algorithm 1 and discuss the model size and time complexity of our model in the following section.

Model Size. For the embedding layer, the size of the vector representation is $m \times d$. Besides, the sequence modeling layer has the parameter of size $k(l+1)d$, and the size of weight matrix is $d \times d$. By adding up all the numbers, the total model size is $2md + 2k(l+1)d + 2 \times 6 \times d^2$. Considering that d and k are small numbers generally less than 100 and the number of layers is usually less than 3, the model size is still a small number.

Time Complexity. In our model, the source of time complexity mainly comes from the process of constructing HIN and random walks. For the HIN, the number of user is m, and the number of item is n. So the computation cost is $O\left(d^2k\right)$. The time complexity for computation through fully connected layer is $O(m^2k)$. The time consumption for modeling node sequences is at least $O\left(kld^2\right)$. As can be seen, our model has extra time expenses in searching for paths of user. Considering the sparsity of the interaction matrices, the compromise is acceptable. After witnessing the improvements presented in Section 4, we insist that the small sacrifice in time complexity is worthy.

4. Experiments

In this section, we conduct experiments on three real-world datasets to evaluate the superiority of proposed model. We focus on answering the following research questions (RQ):

1. **RQ1:** Compared with the traditional recommendation models and the state-of-the-art methods, how does the proposed model perform?
2. **RQ2:** How the key parts of our model affect recommendation performance and whether adversarial learning can improve the accuracy and robustness of the model?
3. **RQ3:** How do some key parameters affect the recommendation performance?
4. **RQ4:** Can the proposed model provide convincingly explanation on recommendation results?

4.1. Experimental settings

Datasets. We use three real-world music datasets: Nowplaying, Xiami, and Yahoo. The Nowplaying is a dataset, which is published by Twitter about users' music listening behavior and contains 87,663 interactions with 8,820 songs. The Xiami dataset contains the listening data of 4,270 users from the Xiami Music APP. Another dataset Yahoo comes from the Yahoo Music APP, which contains some detailed descriptions of music, such as artists, albums, etc. The statistics of the datasets are shown in Table 4. For dataset preprocessing, we follow previous studies [10,14,23,31], and we filter out the user with less than five feedback for three datasets. Besides, we group the interaction records by users and construct HIN, which is used to build node sequences. For each user, we holdout the 80% interaction history to construct the training sets and the rest is for testing.

Evaluation Metrics. To evaluate the recommendation performance of all models, we adopt a leave-one-out evaluation mechanism, similar to the studies in [10,14,31]. We employ two conventional RS evaluation metrics: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). Intuitively, the HR@K metric

Table 1
Summary of related work about HIN-based RS.

Work/study	Approach based on Meta-path	Technique/Method	Limitation
HeteRec [42]	Yes	BPR based on Meta-path	Need to design meta-path for different dataset
HeteSim [26]	Yes	Model based on path similarity	Without considering the weight and choice of the meta-path
OptRank [5]	No	Model using social tagging information as multi-type graph	Without using deep learning to model relationship between nodes
CMF [43]	No	Model using membership relation and MF	Without using deep learning to model social relationship
PathSim [27]	No	Model using A* algorithm to find path	Limited number of path
IF-BPR+ [40]	Yes	Model identifying more informative implicit friends	Need to design meta-path for different dataset
Metapath2vec [3]	Yes	Model using heterogeneous skip-gram model	Without using deep learning to model relationship
HAN [45]	Yes	Model using graph attention network for HIN	Need to design meta-path for different dataset
HetGNN [35]	Yes	Model using graph neural network and node attributes for HIN	Need to design meta-path for different dataset
MEIRec [4]	Yes	Model using graph neural network for HIN	Need to design meta-path for different dataset

Table 2
Summary of related work about knowledge reasoning over graph.

Work/Study	Technique/Method	Limitation(s)
NELLs [28]	Model inferring a new relationship from other learned rules	Without using deep learning to reason
MLN for reasoning [18]	Model learning the weights of a Markov logic network from the HIN	Without using deep learning to reason
MT-KGNN [32]	Model using multi-task neural network to predict relationship	Do not learn complex relationships
Attention-based Multi-hop reasoning [38]	Model using attention mechanism to reason paths between nodes	Without using deep learning to reason
KPRN [36]	Model using LSTM to reason paths between nodes	Without modeling complex relationship

Table 3
Summary of related work about adversarial learning.

Category	Work/ Study	Technique/Method
Using GAN framework to mitigate data noise	IRGAN [34]	Model using GAN to informative retrieval
	RAGAN [41]	Model using GAN to social recommendation
	UGAN [37]	Model using wGAN and cGAN to RS
Using adversarial learning to mitigate data noise	APR [9]	Model adding adversarial perturbations to the model parameters
	AMR [30]	Model using adversarial noise to the images RS

is a commonly used indicator to measure the recall rate while NDCG@K metric is a ranking metric. In this paper, we use $K = 5, 10$ to report HR and NDCG. For both of the evaluation metrics, the higher the value, the better the performance.

Comparison Method. We compare the proposed model with the following models to answer RQ1.

1. **POP**: It is the simplest recommendation model which recommends the most popular songs to users.
2. **BPR** [22]: It adopts a pairwise ranking loss function to optimize the implicit matrix factorization model.
3. **HeteRec** [42]: This model introduces the potential features of different types of meta-paths to represent users and items' connectivity. It designs a global recommendation model that factorizes the user preference matrix to learn the latent representation of users and items and uses the BPR [22] to estimate the proposed model.
4. **HeteroPRS** [26]: This model applies the popularity of items and the user's inherent interest simultaneously. It performs personalized weight learning on various HIN meta-paths and determines the user's intrinsic interest from implicit historical feedback.
5. **CDAE** [39]: It is a Top-N recommender system based on autoencoders, which assumes that observed user-item interactions are the corrupted version of the user's full preference set. This model learns latent representations of corrupted user-item preferences that can better reconstruct users' preferences.
6. **NeuMF** [10]: It is a popular recommendation model based on the neural network to learn users' latent features and items for collaborative filtering. The user embedding and item embedding are then fed into a multi-layer neural architecture, namely neural collaborative filtering layers, to map the latent vectors.
7. **RNN4rec** [13]: It is the first RNN-based recommendation model that uses the N-layer GRU unit to capture the sequential relationship between each node in the sequence. For better parallel computing, this model uses a mini-batch to combat different sessions.
8. **CNN4rec** [31]: It is a model for Top-N sequential recommendation, which regards the different nodes as the "image" among time and latent dimensions to learn sequential patterns using various convolutional filters.
9. **KPRN** [36]: It is a knowledge-aware path recurrent network recommendation model that generates node representation and captures nodes' sequential dependencies using LSTM. Furthermore, this model adopts a new pooling operation to focus on the weights of different paths in the relationship between a user and an item.

Implementation Details. In practical applications, it is not feasible to fully explore all connection paths over HIN. As suggested by previous work [14], we ignore distant connections and adopt fixed-length extraction paths that are efficient for reasoning. For HeteRec model, we use a path similar to the original paper's meta-path [42] to compare with HBLA. Specifically, we design three paths: "user–song—category–song", "user–song–artist–song ", and "user–song–albums–song" to learn the user preference diffusion process. Then, we apply matrix factorization to obtain vector representation of users and songs in the meta-path. Finally, we use optimal parameters to show the recommendation effect. For HeteroPRS, we also use the meta-path designed in the original paper [26], including "song-album-song", "song-artist-song", "song-album-song-artist-song", and "song-artist-song-album-son g". Similarly, we implement the model with parameters that can make the model reach the best.

For fairness, we choose the best performing parameter of all methods as the comparison parameter. For our model, we implement it through TensorFlow, where all parameters are optimally set by grid search. We use Adam [15] to train the model, where the initial learning rate is 0.001, which decreases linearly with the increase of the number of training and the decrease of the loss function. For other parameters, we set the number of layers of the LSTM to 2 and the number of neurons in each layer to 128. The length of each user path is 4 in the Nowplaying dataset, 5 in the Xiami dataset, and 5 in the Yahoo dataset. We experimentally set the dimension of the low-dimensional vector as 32 and the dropout rate as 0.8. All the models are trained based on NVIDIA GeForce GTX 1080 with a batch size of 128. Our code is in https://github.com/0411tony/Yue.

4.2. Overall performance comparison (RQ1)

Table 5 shows the best recommendation performance of all models on three datasets. In particular, the best results in the evaluation metrics are bold.

The POP model, which is the most basic recommendation model, does not make use of historical interaction information between users and items, so it has the worst performance in all datasets. Compared with the POP model, the performance of the BPR is better, but it is not as good as the NeuMF since NeuMF uses neural networks to simulate the complex interaction relationships. From this set of comparative experiments, we can see that considering the interaction information can improve the recommendation performance, and the neural network can better simulate complex interactions.

From the experimental comparison results, we found the performance of HeteRec is lower than other deep-learning based models as it cannot utilize arbitrary meta-paths and uses the clustering of users for recommendation generation. Besides, the parameter

Table 4
The statistics of datasets.

Datasets	Users	Items	Artists	Albums	Density
Nowplaying	155	8,820	1,704	/	6.41%
Xiami	4,270	289,083	32,918	94,969	0.22%
Yahoo	10,732	136,665	20,541	9,440	0.22%

Table 5
Performance comparison of different recommendation models.

Datasets	Metric (%)	POP	BPR	Hete Rec	Hetero PRS	CDAE	NeuMF	RNN 4rec	CNN 4rec	KPRN	HBLA- atten	HBLA	Improve (%)
Now playing	HR@5	3.14	3.39	3.44	3.52	3.48	3.88	4.08	3.99	4.12	4.63	**5.23**	14.18
	HR@10	2.81	3.06	3.23	3.38	3.54	3.96	4.56	4.07	4.11	4.24	**5.11**	20.52
	NDCG@5	0.11	0.12	0.13	0.13	0.12	0.13	0.15	0.13	0.13	0.14	**0.20**	42.86
	NDCG@10	0.09	0.13	0.13	0.14	0.13	0.14	0.16	0.14	0.13	0.14	**0.19**	35.71
Xiami	HR@5	2.12	1.78	2.41	2.53	2.32	2.25	3.21	3.23	3.81	4.56	**5.06**	10.96
	HR@10	1.86	1.50	1.67	1.72	1.91	1.87	2.09	2.41	2.92	3.78	**4.24**	12.17
	NDCG@5	0.05	0.04	0.04	0.04	0.05	0.05	0.07	0.07	0.09	0.13	**0.18**	38.46
	NDCG@10	0.04	0.03	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.11	**0.13**	18.18
Yahoo	HR@5	5.08	8.90	9.23	9.26	9.12	9.34	10.23	11.42	11.88	12.56	**13.73**	9.32
	HR@10	4.62	7.42	7.71	8.07	7.46	8.15	8.73	9.47	10.07	11.64	**12.87**	10.57
	NDCG@5	0.19	0.41	0.37	0.39	0.41	0.42	0.51	0.60	0.73	0.79	**0.85**	7.60
	NDCG@10	0.14	0.24	0.26	0.31	0.28	0.29	0.39	0.41	0.51	0.62	**0.81**	30.65

learning methodology of HeteRec maybe insufficient, which is also a possible reason for the lower recommendation performance. On all the datasets, the model's performance based on the random walk can also achieve the effect of models based on meta-path.

Among the recommendation models which use neural network for reasoning, models using the sequence information between nodes in HIN, such as RNN4rec and CNN4rec, have better performance than NeuMF, which does not take sequences into consideration. Performance improvement is especially obvious on sparse datasets, which demonstrates that considering heterogeneous networks can alleviate the problem of data sparsity. Besides, the performance of CNN-4rec is not as good as that of RNN4rec. This phenomenon may vary due to the limited number of path length in our sequence modeling. LSTM can solve the gradient vanishing problem in learning long-term dependencies. It can better learn the sequence information through the memory function. Compared with the ABLAH model's recommendation performance, RNN4rec is worse, indicating that the bidirectional LSTM can better learn the representation of the node itself. It demonstrates the effectiveness of the attention-based bidirectional LSTM and adversarial learning in the path inference process.

As can be seen from the experimental results, we can conclude that ABLAH performs best among all models on three datasets. There is an increase of 17.28% in HR@10 and 28.12% in NDCG@10 (on average) against the strongest baselines.

4.3. Ablation study (RQ2)

To better understand the impact of each key component of ABLAH on the recommendation performance (attention mechanism (AM)) and reveal the vital role of adversarial learning (adversarial regularization term (ART)), we perform ablation experiments on three datasets. Table 6 shows the results of ABLAH and its variants while keeping the hyper-parameters at their optimal setting. We introduce the variants of ABLAH and analyze their effects as follows:

1. **w/o AM**: It is the variant model of ABLAH that lacks the attention mechanism to reason in HIN. We compare with it to validate the benefits of attention mechanism in the proposed model. The performances show that removing the attention mechanism causes a decline in ABLAH's performance on three datasets. Without the attention mechanism, the importance of each node is considered the same.
2. **ABLH**: It is the basic model of ABLAH that only uses attention-based bidirectional LSTM to reason, without considering the adversarial regularization term. We compare with it to answer the question that how does the adversarial training module affect model performance. The results show that the performance of the ABLAH model is not as good as ABLH when the path length selection is small. Nevertheless, when the path length selected from the HIN becomes longer (i.e., Xiami, and Yahoo), the performance of the ABLAH is gradually better than ABLH. To verify the effect of path length on the two models, we vary the path length of Xiami datasets, as shown in Table 7. We can see that the recommendation model achieves the best results when the number of nodes L is 5. With the increase of L, the performance of the model gradually improves. The performance begins to decline when L exceeds the optimal value, which shows that too many nodes in a path would introduce additional information and more noise, and therefore eventually affect the vector representation of the nodes. However, with the increase of L, our model's performance remains stable without any sharp decline, which shows that our model is robust. Overall, adding the adversarial regularization term can indeed learn more accurate node vector representations and improve the robustness of the embedding representations.
3. **w/o AM and ART**: It is the variant model of ABLAH that only uses bidirectional LSTM to reason in HIN, without considering the attention mechanism and adversarial regularization term. We compare with it to validate the benefits of attention mechanism in the proposed model. We observe that the performance of this model is worse than ABLH and w/o AM, which also verifies the positive effects of attention mechanism and adversarial regularization term on the node representation learning process.

Table 6
Ablation analysis (HR@10) on three datasets. Bold score indicates the best performances.

Models	Datasets		
	Nowplaying	Xiami	Yahoo
w/o AM	5.116	3.393	10.754
ABLH	5.086	3.311	10.460
w/o AM and ART	4.927	3.193	10.231
ABLAH	**5.143**	**3.478**	**10.875**

4.4. Parameter sensitivity analysis (RQ3)

In this section, we study the effect of different settings of parameters on the recommendation performance. When exploring the effect of one hyper-parameter on the performance of the model, we fix other hyper-parameters to the same value.

1. **The dimension of node embedding:** Fig. 3 shows the change of model performance with respect to different dimension number from 16 to 256. It is obvious that model performance even-

Table 7
Performance comparison with respect to different maximum path length L on Xiami dataset. Bold score indicates the best performances.

Models	Metric(%)	2	3	4	5	6
ABLH	HR@10	1.212	2.355	3.140	**3.311**	3.183
	NDCG@10	0.047	0.091	0.123	**0.128**	0.125
ABLAH	HR@10	2.177	2.591	3.115	**3.478**	3.229
	NDCG@10	0.091	0.114	0.125	**0.134**	0.128

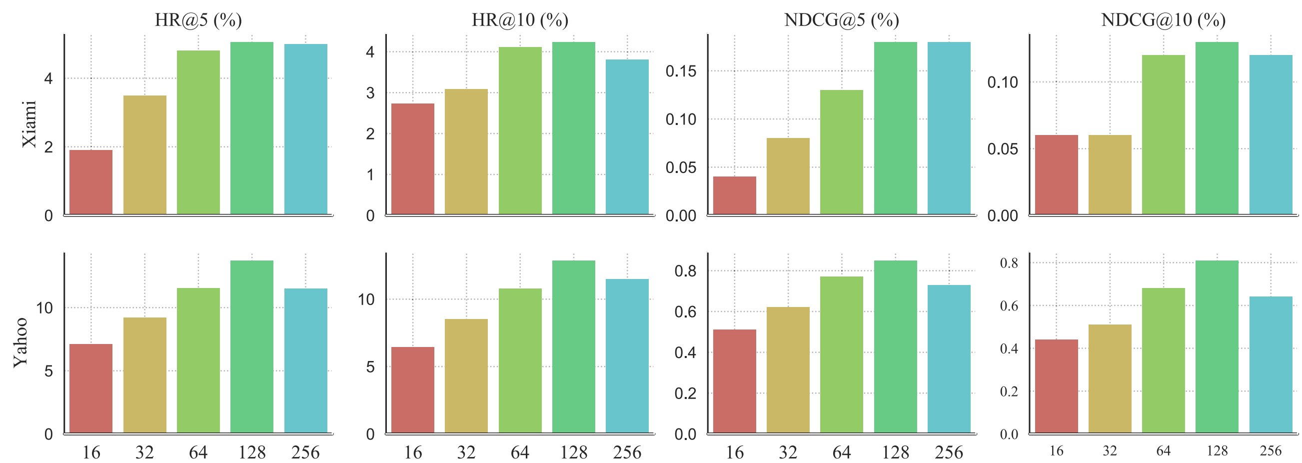

Fig. 3. Effect of node embedding dimension d on model.

tually converges with the rise of dimension. Larger embedding dimensions do not yield better model performance, especially in sparse datasets. Concretely, the optimal embedding dimensions is 128. Therefore, we set the dimension d to 128 in other experiments.

2. **The depth of neural network layers:** In our model, we adopt neural networks to project the final state. Hence, the depth of the neural network is an important parameter. The optimal layer number of neural network is searched in 1, 2, 3, 4, 5 in our experiment. We present the results in Fig. 4(a). Through the experimental results, we can find that when the layer of the neural network is 4, the model achieves better performance in the Nowplaying dataset. Using 2 layers of neural networks yields the best performance in Xiami and Yahoo datasets, indicating that the sparser the data, the deeper the neural network layers should be.
3. **The number of neurons:** Besides, we analyze the number of neurons in each layer of the neural network, and the number of neurons in each layer is tuned among 16, 32, 64, 128, 256. As shown in Fig. 4(b), when the number of neurons is 64, our model achieves the best performance. The HR@10 increases first and then decreases with the number of neurons, so we choose 128 as the number of neurons for the Yahoo and Xiami dataset.
4. **The value of ϵ:** Next, we set the dimension of node embedding as 128 and tune the value of ϵ, which is used to control the regularization term. We examine how it influences the model performance, when it varies from 0.1 to 1.0. As shown in Fig. 4(c), the optimal performance is obtained when ϵ is near 0.4 in the Nowplaying dataset. When the value of ϵ is 0.6, our model achieves the best performance in the Xiami and Yahoo dataset.
5. **The dropout rate:** Finally, we study the influence of dropout rate on model performance. Fig. 4(d) shows the results with respect to different dropout rate from 0.1 to 0.9 on three datasets. The Nowplaying dataset prefers a small value of 0.3. For the more sparse dataset (e.g., Xiami and Yahoo), the best performances are achieved when dropout rate is set to larger value, concretely 0.5 and 0.6 respectively. We can also see that the recommendation performance gradually decreases when the dropout rate becomes larger. We reckon that the model may suffer from under-fitting when the drop rate is too large.
6. **The number of nodes:** We show the effect on the accuracy of the number of different nodes in the path (that is, different path lengths) on three datasets. It can be seen from Fig. 4(e) that the recommendation accuracy first increases and then decreases as the number of nodes increases. When the number of nodes is around 5, the accuracy reaches the peak. The reasons for this trend may be too many nodes introduce more noise, and too short path length fail to learn the sequence relationship of nodes in the network. In different datasets, the optimal path length may be different.
7. **The number of paths:** On three datasets, we show the effect of different random walk times on accuracy. It can be seen from Fig. 4(f) that the recommendation accuracy gradually decreases as the number of paths. When the number of paths is around 10, the recommendation performance reaches the optimal on three datasets. Besides, when the number of paths exceeds 25, the performance drops sharply, possibly because too many paths include too many paths of the same path-type, resulting in model overfitting. Another possible reason is that too many paths introduce too much noise. Moreover, there is no doubt that too few paths fail to learn the sequence relationship of nodes in the network. In different datasets, the optimal number of paths may be different.

4.5. Case studies (RQ4)

It is innovative that our model uses attention-based bidirectional LSTM to infer the node sequences over HIN. To demonstrate the feasibility of our model on explainable recommendation task, we randomly select a user $User_1$ from the Xiami dataset and show its 4 paths to song $Song_4$. As shown in Fig. 5(a), we can see that

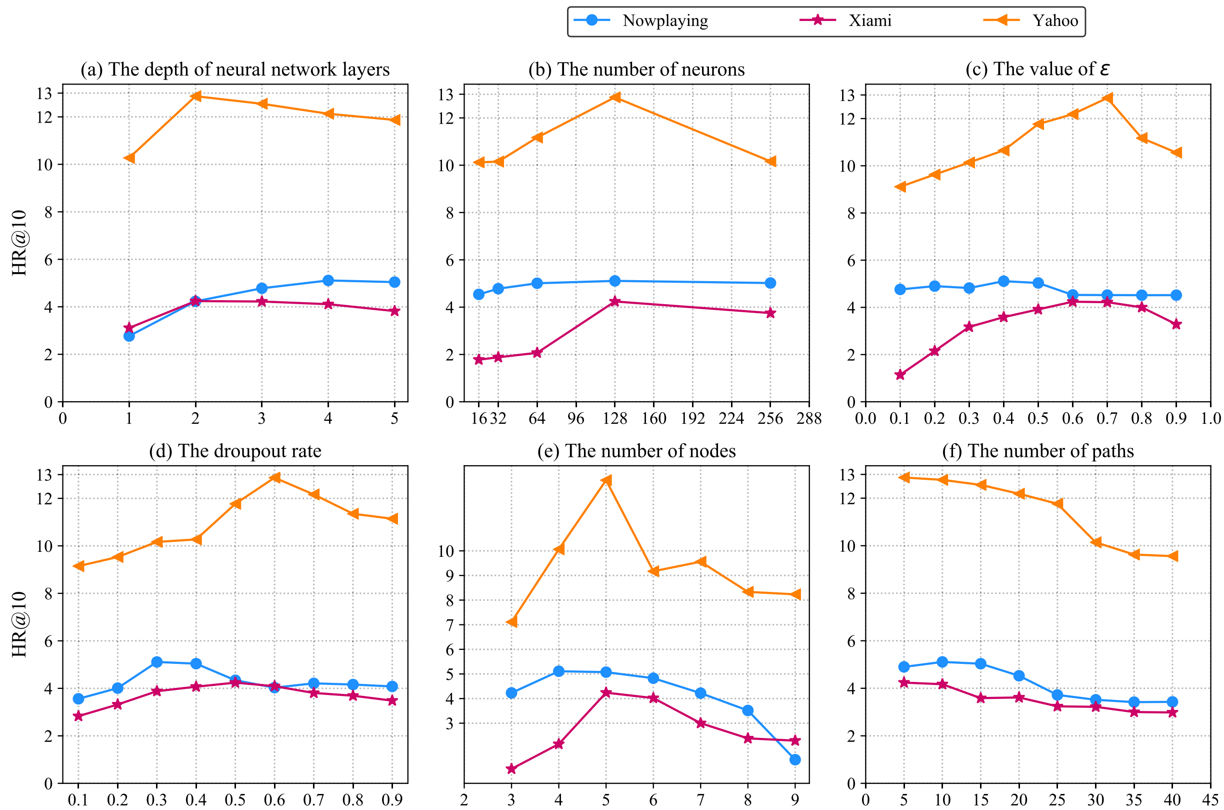

Fig. 4. Effect of varying parameters on three datasets with HR@10.

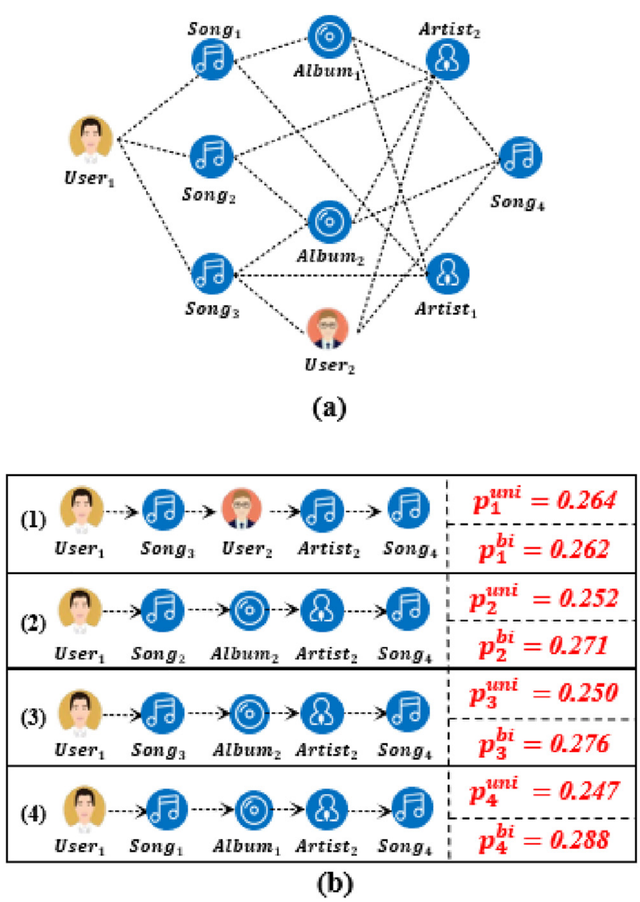

Fig. 5. Visualization of four paths with prediction scores for the user of $User_1$ in Xiami dataset. The prediction scores are normalized for illustration.

$Song_4$ is related to $Album_2$, $Artist_2$, and $User_2$. Through the display of different paths, we find that the paths describes the connectivity between $User_1$ and $Song_4$ from different perspectives, which can be seen as the reason why songs are recommended to users. Based on this, the target song is more persuasive when it is recommended to the user.

As shown in Fig. 5(b), we select 4 paths from the HIN of Xiami and calculate the weights of different paths from two ways: bidirectional and unidirectional. The weight of the path calculated by bidirectional is generally higher than that calculated by unidirectional, and the path (4) has the highest probability. Therefore, when the model recommends the $Song_4$ to the $User_1$, the reason for the recommendation can be shown to the user at the same time. For example, the user, who have listened $Song_1$, may be also interested in the $Song_4$, which belongs to the same album and same singer as $Song_1$. However, if we use unidirectional method, the model will recommend $Song_4$ to $User_1$ according to path (1), which shows that considering the bidirectional relationships of the nodes could better capture the complex nodes sequences.

5. Conclusion & future work

In this paper, we introduce an attention-based bidirectional LSTM with adversarial learning for path-based reasoning over HIN, named ABLAH, to capture the complex node sequence information and highlight the different importance of nodes in HIN. As for noisy relationships in the node sequence, we use the adversarial regularization term to learn more accurate node vector representations and improve the robustness of the embedding representations. Extensive experimental results on three real-world datasets show that the superiority of our model compared to other state-of-art baselines. The case study validates the feasibility of the proposed model on explainable recommendation task.

There are still several directions to be explored in the future. A valuable one is to consider the position information of nodes in the sequence, instead of merely simulating the sequence of paths through the model. Another interesting direction is to consider the multiple types of edges and different attributes of nodes in HIN.

CRediT authorship contribution statement

Junwei Zhang: Investigation, Writing - original draft, Writing - review & editing. **Min Gao:** Conceptualization, Validation, Writing - review & editing. **Junliang Yu:** Validation, Resources, Writing - review & editing. **Linda Yang:** Validation, Writing - review & editing. **Zongwei Wang:** Project administration, Funding acquisition. **Qingyu Xiong:** Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (2019YFB1706101), the Natural Science Foundation of Chongqing, China (cstc20 20jcyj-msxmX0690), the Key Research Program of Chongqing Technology Innovation and Application Development (cstc2019jscx-fxydX0012), and the Overseas Returnees Innovation and Entrepreneurship Support Program of Chongqing (cx2020097), and the Fundamental Research Funds for the Central Universities of Chongqing University (2020CDJ-LHZZ-039).

References

[1] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, J. Tang, Representation learning for attributed multiplex heterogeneous network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019 (2019) 1358–1368.

[2] T. Chen, Y. Sun, Task-guided and path-augmented heterogeneous network embedding for author identification, Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017 (2017) 295–304.

[3] Y. Dong, N.V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2017 (2017) 135–144.

[4] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, Y. Li, Metapath-guided heterogeneous graph neural network for intent recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019 (2019) 2478–2486.

[5] W. Feng J. Wang, Incorporating heterogeneous information for personalized tag recommendation in social tagging systems, in: The 18th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2012 (2012) 1276–1284.

[6] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2016 (2016) 855–864.

[7] He R., Kang W., McAuley J.J, Translation-based recommendation, Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017 (2017) 161–169.

[8] R. He, J.J. McAuley, Fusing similarity models with markov chains for sparse sequential recommendation, in: IEEE 16th International Conference on Data Mining, ICDM 2016 (2016) 191–200.

[9] X. He, Z. He, X. Du, T. Chua, Adversarial personalized ranking for recommendation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018 (2018) 355–364.

[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, WWW 2017 (2017) 173–182.

[11] Hidasi B., Karatzoglou A., Recurrent neural networks with top-k gains for session-based recommendations, CoRR (2017), abs/1706.03847.

[12] B. Hidasi, A. Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018 (2018) 843–852.

[13] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, in: 4th International Conference on Learning Representations, ICLR 2016 (2016) 153–162.

[14] Kang W., McAuley J.J, Self-attentive sequential recommendation, IEEE International Conference on Data Mining, ICDM 2018 (2018) 197–206.

[15] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 (2015) 153–162.

[16] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, L. Getoor, Personalized explanations for hybrid recommender systems, in: Proceedings of the 24th International Conference on Intelligent User Interfaces, IUI 2019 (2019) 379–390.

[17] J. Kunkel, T. Donkers, L. Michael, C. Barbu, J. Ziegler, Let me explain: Impact of personal and impersonal explanations on trust in recommender systems, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019 (2019) 487.

[18] Kuzelka O., Davis J., Markov logic networks for knowledge base completion: A theoretical analysis under the MCAR assumption, in: Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, AUAI Press, 2019, p. 427.

[19] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, (2017), pp. 1419–1428.

[20] Lin Y., Liu Z., Sun M., Liu Y., Zhu X., Learning entity and relation embeddings for knowledge graph completion, Proceedings of the Twenty-Ninth Conference on Artificial Intelligence, AAAI 2015 (2015) 2181–2187.

[21] Q. Liu, Y. Zeng, R. Mokhosi, H. Zhang, STAMP: short-term attention/memory priority model for session-based recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, (2018), pp. 1831–1839.

[22] Rendle S., Freudenthaler C., Gantner Z., Schmidt-Thieme L., BPR: bayesian personalized ranking from implicit feedback, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI (2009) 452–461.

[23] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in: Proceedings of the 19th International Conference on World Wide Web, WWW 2010 (2010) 811–820.

[24] M.T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you?: Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016 (2016) 1135–1144.

[25] C. Shi, B. Hu, W.X. Zhao, P.S. Yu, Heterogeneous information network embedding for recommendation, IEEE Trans. Knowl. Data Eng. 31 (2019) 357–370.

[26] C. Shi, X. Kong, Y. Huang, P.S. Yu, B. Wu, Hetesim: A general framework for relevance measure in heterogeneous networks, IEEE Trans. Knowl. Data Eng. 26 (2014) 2479–2492.

[27] Y. Sun, J. Han, X. Yan, P.S. Yu, T. Wu, Pathsim: Meta path-based top-k similarity search in heterogeneous information networks, PVLDB 4 (2011) 992–1003.

[28] D.A. Talbert, W. Eberle, M. Liu, Never-ending medical learning, in: AMIA 2016, American Medical Informatics Association Annual Symposium, AMIA 2016 (2016) 0–0.

[29] Y.K. Tan, X. Xu, Y. Liu, Improved recurrent neural networks for session-based recommendations, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016 (2016) 17–22.

[30] Tang J., He X., Du X., Yuan F., Tian Q., Chua T., Adversarial training towards robust multimedia recommender system, CoRR 2018 (2018), abs/1809.07062.

[31] Tang J., Wang K., Personalized top-n sequential recommendation via convolutional sequence embedding, in: 2018 Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, WSDM, 2018, pp. 565–573.

[32] Y. Tay, L.A. Tuan, M.C. Phan, S.C. Hui, Multi-task neural network for non-discrete attribute prediction in knowledge graphs pp, (2017) 1029–1038.

[33] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I., Attention is all you need, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, NIPS, 2017, pp. 5998–6008.

[34] J. Wang, L. Yu, W. Zhang, Y. Gong., Y. Xu, B. Wang, P. Zhang, D. Zhang, IRGAN: A minimax game for unifying generative and discriminative information retrieval models, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017 (2017) 515–524.

[35] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: Proceedings of the 2019 World Wide Web Conference on World Wide Web, WWW 2019 (2019) 2022–2032.

[36] Wang X., Wang D., Xu C., He X., Cao Y., Chua T., Explainable reasoning over knowledge graphs for recommendation, in: 2019 The Thirty-Third AAAI Conference on Artificial Intelligence, 2019, AAAI, 2019, pp. 5329–5336.

[37] Z. Wang, M. Gao, X. Wang, J. Yu, J. Wen, Q. Xiong, A minimax game for generative and discriminative sample models for recommendation, in: Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD, 2019, (2019, pp. 420–431.

[38] Z. Wang, L. Li, D.D. Zeng, Y. Chen, Attention-based multi-hop reasoning for knowledge graph, in: 2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018 (2018) 211–213.

[39] Wu Y., DuBois C., X.Zheng A., Ester M., Collaborative denoising auto-encoders for top-n recommender systems, Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, ICDM 2016 (2016) 153–162.
[40] J. Yu, M. Gao, J. Li, H. Yin, H. Liu, Adaptive implicit friends identification over heterogeneous network for social recommendation, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018 (2018) 357–366.
[41] J. Yu, M. Gao, H. Yin, J. Li, C. Gao, Q. Wang, Q., 2019. Generating reliable friends via adversarial training to improve social recommendation, in: 2019 IEEE International Conference on Data Mining, ICDM 2019 (2019) 768–777.
[42] Yu X., Ren X., Sun Y., Gu Q., Sturt B., Khandelwal U., Norick B., Han J., Personalized entity recommendation: a heterogeneous information network approach, in: Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, 2014, pp. 283–292.
[43] Q. Yuan, L. Chen, S. Zhao, Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation (2011) 245–252.
[44] W. Yuan, H. Wang, X. Yu, N. Liu, Z. Li, Attention-based context-aware sequential recommendation model, Inf. Sci. 510 (2020) 122–134.
[45] C. Zhang, D. Song, C. Huang, A. Swami, N.V. Chawla, N.V., 2019. Heterogeneous graph neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019 (2019) 793–803.
[46] Zhang S., Tay Y., Yao L., Sun A., Next item recommendation with self-attention, CoRR 2018 (2018), abs/1808.06414.
[47] Zhang Y., Chen X., Explainable recommendation: A survey and new perspectives, CoRR 2018 (2018), abs/1804.11192.
[48] T. Zhao, J.J. McAuley, I. King, Leveraging social connections to improve personalized ranking for collaborative filtering, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014 (2014) 261–270.

Junwei Zhang received the B.S. degree from Chongqing University, Chongqing, China, in 2018, where he is currently pursuing the master's degree with the School of Big Data & Software Engineering, Chongqing University. His research interests include recommender systems and anomaly detection.

Min Gao received the MS and PhD degrees in computer science from Chongqing University in 2005 and 2010 respectively. She is an associate professor at the School of Big Data & Software Engineering, Chongqing University. She was a visiting researcher at University of Reading and Arizona State University. Her research interests include recommendation systems, service computing, and data mining. She has published over 40 refereed journal and conference papers in these areas. She has grants from the National Natural Science Foundation of China, the China Postdoctoral Science Foundation, and the China Fundamental Research Funds for the Central Universities. She is member of the IEEE and CCF.

Junliang Yu received the B.S. and MS degree in Software Engineering from Chongqing University, Chongqing, China. Currently, he is a Ph.D. student with the school of Information Technology and Electrical Engineering at the University of Queensland, Queensland, Australia. His research interests include recommender systems and anomaly detection.

Linda Yang received the B.E. and M.S. degrees in computer software, and the Ph.D. degree in computer science from Peking University, China, in 1993, 1996, and 1999, respectively. She was a Lecturer with Robert Gordon University and a Senior Research Fellow of the University of St. Andrews, U.K., and Cardiff University, U.K. She is currently a Senior Lecturer with the University of Portsmouth, U.K. She has worked on U.K. and European funded research projects and has published over 50 papers in refereed journals and conferences. Her research interests include recommendation systems, information retrieval, and data mining as well as their applications in social networks, health informatics, and business. She is a member of the IET, WES, and HEA.

Zongwei Wang received the B.S. degree from Chongqing University, Chongqing, China, in 2018, where he is currently pursuing the master's degree with the School of Big Data & Software Engineering, Chongqing University. His research interests include recommender systems and anomaly detection.

Qingyu Xiong received the B.S. and M.S. degree from the School of Automation, Chongqing University, in 1986 and 1991, respectively, and the Ph.D. degree from the Kyushu University of Japan in 2002. He is currently the Dean of the School of Big Data & Software Engineering, Chongqing University. His research interests include neural networks and their applications. He has authored over 100 journal and conference papers in these areas. He has over 20 research and applied grants.