Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07490-0

ORIGINAL ARTICLE

Feature-enhanced embedding learning for heterogeneous collaborative filtering

Wenchuan Yang[1] · Jichao Li[1] · Suoyi Tan[1] · Yuejin Tan[1] · Xin Lu[1]

Received: 22 December 2021 / Accepted: 27 May 2022

Abstract
Heterogeneous information network (HIN) has recently been receiving increasing attention in recommender systems due to its practicability in depicting data heterogeneity. The rich structural and semantic information embodied in the HIN can help mining latent features of users and items for recommendations. However, almost all existing HIN-based recommendation methods focus on the design of complicated learning architecture while using simply initialized features. In this paper, we propose a novel feature-enhanced embedding learning model which combines informative feature initialization strategy with simple learning architecture for heterogeneous collaborative filtering. We first build multiple homogeneous sub-networks by extracting different relations guided by meta-paths from the HIN. We then design a comprehensive feature initialization strategy that contains *semantic* and *spatial encoding* module to characterize the node feature. After that, a simple learning architecture based on multi-layer perceptron is applied to learn the latent representation of users and items. Next, a novel convolutional neural network-based fusion mechanism is used to determine the attention weight of semantic relations and compress multiple embedding vectors into a compact representation to apply for final recommendation. Finally, we conduct extensive experiments on two classic datasets to demonstrate the effectiveness and feasibility of the proposed FHetCF method in solving HIN-based recommendation tasks. Results show that the proposed method soundly outperforms the competitive baselines by 1.71 to 10.46% on hit ratio and 3.17 to 13.75% on normalized discounted cumulative gain, respectively. The proposed method opens up a new avenue to effectively utilize heterogeneous information to improve recommendation performance.

Keywords Heterogeneous information network · Collaborative filtering · Feature initialization strategy

1 Introduction

Due to the rapid growth of online information, it has become difficult for online users to quickly find items or services of interest from the massive amounts of available information. Recommender systems have been developed as an effective solution to deal with information overload issue and help users to filter redundant information. They have been widely applied in many fields, including e-commerce, advertising, education, and so on [1–7]. The general idea is for recommender systems to predict a user's preference of items that have not yet been purchased and recommend items that cater to their needs or taste. Collaborative filtering (CF) is a typical approach used by recommender systems which predicts the user's preference based on historical user–item interactions [8, 9]. Generally, collaborative filtering methods comprise two important components: (1) *latent factor learning*, which translates the user and item into vectorized representations, and (2) *interaction modeling*, which predicts the interaction between users and items using latent representation. For example, matrix factorization (MF) [10], which is one of the sought-after CF methods, projects users and items into a low-dimensional space and obtains their latent factors. The user–item interaction is then modeled using the inner product of latent factors.

However, a major limitation of CF methods is that they only exploit direct user–item interaction data and neglect

✉ Xin Lu
xin_lyu@sina.com

[1] College of Systems Engineering, National University of Defense Technology, Changsha, Hunan 410073, People's Republic of China

Published online: 22 June 2022

different types of dependencies among users and items, so they cannot completely describe users personalized interests and the characteristics of items [11, 12], and the performance is usually restricted due to data sparsity. Due to the practicability of depicting data heterogeneity, heterogeneous information network (HIN) has been verified as an effective tool to incorporate diverse dependencies into recommendation models [13–15], which have recently been attracting increasing attention. A HIN contains multi-typed objects connected by edges belonging to multiple types, providing a feasible way to organize complex relations between users and items in recommender systems [16–18]. In a HIN-based recommender system, the main task is to generate users' and items' presentations for recommendation by capturing heterogeneous information.

In recent years, some attempts have been made to utilize HIN in recommender systems and have obtained encouraging performance improvements. The basic idea of most existing HIN-based recommendation models is to use the interactive structural information like meta-paths to guide the representation learning process. However, these methods still face two problems. First, it is usually time-consuming to generate sufficient meta-path instances (or meta-graphs) for representation learning with HIN [18, 19], Limiting its applications for large for large network in real-world scenarios. Second, the learning module designed for capturing various interactive information embedded in the HIN is sophisticated, which will degrade the efficiency and hinder the real-world application [49, 50].

Based on the above analysis, we will address the following challenges which are not well-solved by existing works on HIN-based recommender system in this study:

(1) **How to design a simple but effective mechanism for learning latent representations based on the HIN.** Recent research [27] argues that a careful design of a neighborhood selection method could improve the recommendation performance effectively even with a single-layer graph convolutional network (GCN). This inspired us to rethink the architecture of a HIN-based recommendation method. Instead of employing meta-paths to guide learning process, we focus on the design of feature initialization strategy which aimed to better encoding the node feature fed into the learning module. Existing feature initialization strategies, such as random initialization or path count [2, 28, 29] which are commonly used in other machine learning tasks, are not considered to be feasible solutions under the heterogeneous network setting since they neglect or only capture part of the information embodied in HINs. Although some attempts utilize extra context features of entities i.e., research keywords for authors [43], this kind of information is not available most of the time. Thus, a well-designed initialization module to extract the informative feature in the HIN itself is required.

(2) **How to recognize the importance of different semantic relations.** Semantic relations in fact reflect the various factors that may contribute to decision-making process. Thus, to predict the interaction probability between users and items, the recommendation model should be capable of fusing latent representation generated from different semantic relations effectively. Existing methods such as averaging or MLP-based attention are not feasible options in the heterogeneous setting since they cannot fully distinguish the importance among different semantic relations (we will show this in the Ablation Study section). A proper fusing mechanism to aggregate the diverse semantic information is always expected.

To address these challenges, we propose a **F**eature-enhanced embedding model for **Het**erogeneous **C**ollaborative **F**iltering (FHetCF). We first employ meta-path [30], a composite relation sequence, to build the corresponding user–user/item–item homogeneous network. Thus, we obtain a collection of homogeneous sub-networks by decomposing the HIN with different semantic relations. Then we propose a feature-enhanced embedding learning method to learn latent representation for users and items. As feature initialization strategy is our main focus in the paper, we propose *semantic* and *spatial encoding* to capture semantic and geometric similarity between nodes and characterize the node feature in the input layer of the neural embedding module. For clarity, the embedding module is utilized to map the node feature into low-dimensional space and learn the latent representation of users and items. Furthermore, to automatically determine the importance of semantic relations for each user (or item), we leverage the power of one-dimensional convolutional neural network (1D-CNN) [31] and design an attention mechanism to capture personalized preference information. The contribution of the paper is summarized as follows:

- We design a feature-enhanced embedding model that infers the latent representation of users and items based on the semantic and spatial encoding technique. Empirical study demonstrates that an informative feature initialization with a simple embedding learning architecture is capable of obtaining satisfactory recommendation performance.
- We propose a fusion mechanism based on a 1D-CNN to aggregate latent factors from different semantic relations. To the best of our knowledge, this is the first work which utilizes 1D-CNN to learn personalized weight on meta-paths.
- We provide a comprehensive analysis on common feature initialization strategies in existing research and investigate their recommendation performance in our experiment.

The rest of the paper is organized as follows. Section 2 provides an overview of related work. Section 3 introduces basic concepts and defines the top-N recommendation problem. The detail of our model is presented in Sect. 4. Section 5 presents the settings of experiments. The experimental results are analyzed in Sect. 6. Finally, we summarize our work and discuss future work in Sect. 7.

2 Literature review

2.1 Representation learning on HIN

A HIN defines a network containing various types of nodes and links, and it is a popular technique for modeling complex relations between different types of entity in the real world. Representation learning on HINs projects nodes in a HIN to a low-dimensional vector space, and the goal is to preserve the original geometric structure of a HIN by optimizing the vectorized node representation [32]. The obtained embeddings served as the input feature for a variety of downstream machine learning tasks such as node classification [33, 34], community detection [35], and clustering [36].

A significant line of research has been addressed for representing heterogeneous information networks. For example, Metapath2vec [39], employs random walks [38] guided by meta-path on nodes and uses a skip-gram model to learn network structure. HIN2Vec [40] uses relations specified in forms of meta-paths as the prediction target to refine node embedding. Unlike Metapath2vec, the training tuple is generated by homogeneous random walks regardless of node types and edge types in HIN2Vec. Instead of performing random walks on HINs, PTE [41] decomposes the HIN into a set of bipartite graphs, with each representing one relation type. The learning objective of PTE is obtained by jointly solving second-order proximity over all bipartite graphs. In recent years, graph neural network (GNN) based methods have become a trending approach to facilitate representation learning in graph data. The embedding learning process can be seen as the aggregation of neighborhood information. R-GCN [42] utilizes the relational graph convolutional neural network (GCN) to learn the latent representation, which multiple convolution matrices are used to embed the heterogeneity of relations in the HIN. HGAT [43] adopts the graph attention network to perform embedding learning process, and the model aggregates the information by assigning personalized weight for neighbors came from different hops and meta-paths. MAGNN [44] focuses on the content information in meta-path instances and introduced a GNN learning module to enrich the embeddings by aggregating node features along the instance. Later, Hu et al. [45] propose HGT which employs transformer-based attention to learn the type-specific representations of nodes and links, so that informative meta-paths could be extracted automatically rather than manual design. Similarly, HetSANN [46] designs a meta-path free learning method which contains multiple graph attention layers aimed for encoding various types of relation in the HIN. Recently, Yu et al. [47] design R-HGNN which adds a relation representation learning module to guide the relation-specific node representation learning process. Zhao et al. [48] extend the graph structure learning method to the heterogeneous network setting, and proposed a framework named HGSL which simultaneously learns the HIN structure and GNN parameters.

To conclude, these methods have shown encouraging performance in many downstream machine learning tasks for HINs. Since the main objective of a recommender system is to generate a comprehensive vectorized representation for users and items, HIN-based embedding methods can provide guidance on utilizing heterogeneous information to improve recommendation performance.

2.2 HIN-based recommender system

Collaborative filtering is a popular approach in the recommender system. It makes predictions of a user's preference based on preference information collected from many other similar users. Matrix factorization (MF) [10] treats the user–item interaction as the inner product of the latent factors of users and items. Based on MF, BPR [51] introduces a pairwise ranking objective function to optimize prediction performance. Since the linear inner product is not capable of capturing complicated intrinsic relations between users and items, NeuMF [52] leverages the power of a neural network to model the user–item interaction. Recently, graph-based CF methods have gained increasing attention due to their ability for high-order connectivity. NGCF [53] transforms user–item interactions to bipartite graphs and uses graph convolutional operation to generate embedding of the user and item.

Notably, the data sparsity issue is commonly observed in real recommender systems [54], which could largely limit the effectiveness of CF methods. To tackle this issue, a number of approaches that leverage auxiliary information to generate better representations have been proposed in recent years. One of such approaches is Cross-domain collaborative filtering (CDCF) [55], which aims to exploit the knowledge from auxiliary domains to improve the performance in the target domain. Yu et al. [56] design a two-side transfer method which adopts Funk-SVD [10] to extract informative features from auxiliary domains, and then users' and items' features could be effectively enriched in the target domain. TSSEAE [57] addresses the domain selection problem in an ensemble learning manner and utilizes Pareto Ensemble Pruning technique [58] to determine the optimal combination of auxiliary domains.

As a burgeoning direction, HIN has been proven to be a powerful tool for characterizing diverse entities and complex relations in practical recommendation scenarios. There is a surge of works on building an efficient recommender system based on HIN. For instance, HERec [19] utilizes meta-path-based random walks to derive homogeneous node sequences from the HIN, and it adopts deepwalk [38] to generate node embedding for recommendations. LGRec [20] combines local neighbor information aggregation and meta-path-based interaction prediction to learn the comprehensive representation of users and items. MCRec [18] introduces meta-path-based context to enhance interaction modeling. NeuACF [11] adopts various meta-paths to extract indirect interaction information embodied in the HIN, after which the information is fed into a neural network-based model to make recommendations. As inaccurate information extraction may be caused due to the separate modeling of users and items in meta-paths, an embedding model HueRec [28] has been proposed to map users and items into a unified embedding space based on the common characteristics of all meta-paths. Recently, HetNERec [21] transforms the original HIN into multiple heterogenous co-occurrence networks and designs a regularized matrix factorization model to perform the recommendation. Jin et al. [22] propose GraphHINGE to enhance the latent representations of nodes in the HIN through modeling the interaction information of their intra-metapath and inter-metapath neighborhood based on a convolutional framework. HAF [23] utilizes meta-graph to capture similarity between users and items guided by various semantics and employs a "matrix factorization + factorization machine" framework to perform the feature fusion for the recommendation. Later, Xie et al. [24] add an attentive Bi-LSTM module to learn the embeddings for meta-graph instances and further incorporated into the rating prediction.

In summary, the design of existing models can be seen as a combination of simply initialized features and complex learning architecture. For clarity, "Simply" refers to characterizing the node feature by less-informative methods, i.e., standard normal initialization, Xavier initialization, or meta-path-based features. "Complex" means that the learning architecture consists of multiple components or sophisticated design; e.g., a parallel learning design that neighborhood-based recommendation is combined with a meta-path-based interaction prediction model [20, 28] or a hierarchical aggregation design that object-level aggregation is followed by type-level aggregation [59]. However, the potential of combining carefully designed feature initialization and relatively simple learning mechanism has not been explored so far since an effective initialization strategy is also capable of capturing the abundant information embodied in the HIN. Besides, semantic relations represent different motivations behind interaction behavior, so it is necessary to design a method to learn the importance of various motivations for an accurate prediction. These issues have motivated us to explore the design of embedding a learning algorithm and information fusion mechanism in the recommendation model with the HIN.

3 Problem definition

In this section, we begin with an introduction of notations and basic concepts used in our model. For convenience, we list some symbols and their descriptions in Table 1. Then we discuss the definition of top-N recommendation.

Definition 1 A *heterogeneous information network* [37] is defined as a graph $G = (V,E)$ that is composed of a set of nodes V with the corresponding edge set E. Here, a node type mapping function $f_n : v \rightarrow o$ and an edge type mapping function $f_e : e \rightarrow r$ define the type of nodes and edges. O and R are the set of node types and edge types, respectively, satisfying $|o| + |r| > 2$.

Definition 2 A *meta-path* [30] is a node sequence connected by multiple relations that portrays specific semantic relations between entities. Formally, a meta-path p can be written as $p = \tau_1 \xrightarrow{r_1} \tau_2 \xrightarrow{r_2} \cdots \xrightarrow{r_{l-1}} \tau_l$, with τ_i representing the type of node i.

In Fig. 1(a), we present an example of HIN which consists of multi-type entities (e.g., Users (U), Items (I) and Brands (B)) and various interaction relations. Specifically, we are interested in meta-paths, for example, the meta-path *User–Item–User* (U–I–U) shows that two users both bought the same item, and the meta-path *User–Item–Brand–Item–User* (U–I–B–I–U) implies that two users both purchased items belonging to the same brand.

Definition 3 The *commuting matrix* is calculated based on meta-path. Given a meta-path $P = \tau_1\tau_2 \cdots \tau_l$,$A_{\tau_i\tau_j}$ is the adjacency matrix between type τ_i and τ_j, the commuting matrix is defined as $C = A_{\tau_1\tau_2} \bullet A_{\tau_2\tau_3} \bullet \ldots \bullet A_{\tau_{l-1}\tau_l}$.

Heterogeneous collaborative filtering. In this study, the recommendation problem is translated into predicting the probability of user–item interaction with HIN. Given the heterogeneous information network G and user–item interaction matrix R, the goal is to generate a ranked item list for each user based on predicted interaction probability. The implicit user–item interaction matrix R can be defined as:

Table 1 Commonly used notations in the paper

Notations	Descriptions
$G_i = (V, E)$	A network/graph
A	Adjacency matrix
C	Commuting matrix
$P = \{p_1, p_2, \ldots, p_t\}$	A set of meta-paths
V_u	A set of users
V_i	A set of items
u, v	Number of users and items
n	Number of nodes in a network
$R \in \mathbb{R}^{u \times v}$	User-item interaction matrix
u_i	User i
v_j	Item j
τ_k	The type of node in the HIN
r_k	Relation between two types of nodes in the HIN
α_k	The attention weight
nd_i	Node i in the HIN
e_{ij}	An edge between node i and j
w_{ij}	The weight of edge e_{ij}
$\boldsymbol{s}_i$	Latent representation of node i
$\boldsymbol{W}^i \boldsymbol{b}^i$	Learnable convolution kernel and bias of ith layer in CNN
$\boldsymbol{W}_i, \boldsymbol{b}_i$	Learnable weight matrix and bias of ith layer in MLP
$\boldsymbol{E}_i^p, \boldsymbol{e}_i^p$	A latent variable matrix, vector based on meta-path p
$\boldsymbol{L}$	Loss function
ϑ	Learnable model parameters
$exp()$	Exponentiation
$\sigma(\bullet)$	Some activation functions
$*$	Convolution operation

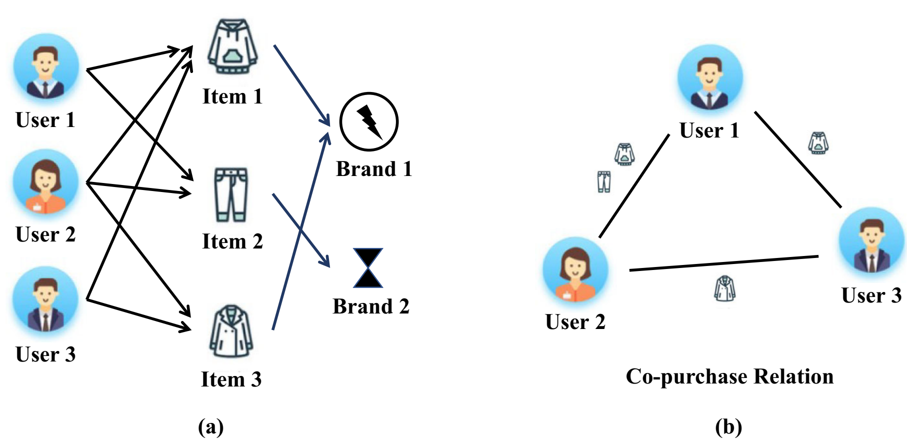


Fig. 1 **a** A HIN contains various entities (users, items, brands) and the corresponding relationships between the entities. **b** An example of homogeneous network extracted by *User–Item–User* meta-path

$$r_{u,v} = \begin{cases} 1, & ifuseruinteractedwithitemv \\ 0, & otherwise \end{cases} \quad (1)$$

where m and n denote the number of users and items, respectively. Then the user–item interaction with various side information can be modeled as a corresponding HIN $G = (V, E)$, where $V = V_u \cup V_i$, and V_u, V_i stand for the set of users and items, respectively.

4 Methodology

The main idea behind our methodology is the design of a FHetCF framework that involves a feature-enhanced learning module to fully exploit the different information embodied in HINs and a fusion mechanism to integrate information for recommendation prediction. The overall structure is shown in Fig. 2. First, based on the constructed HIN, we use multiple symmetric meta-paths to capture semantic relations between users (or items) and obtain a set of user–user (or item–item) sub-networks (Fig. 2 (1)). Second, we introduce an embedding learning module to generate latent representations for users and items. Particularly, a feature initialization strategy is proposed to encode spatial and semantic relation into node features, and an MLP model is built for consecutive embedding learning (Fig. 2 (2)). After that, we aggregate information through a well-crafted attention mechanism at path level (Fig. 2 (3)). Finally, the model obtains the final prediction (Fig. 2 (4)). The details of the architecture are illustrated in the following subsections.

4.1 Homogeneous Sub-network Construction

Heterogeneous nodes and links in the HIN provide rich auxiliary information on users and items that reveals the user's implicit preference (or item's hidden characteristics). For example, as depicted in Fig. 1(a), User 1 and User 3 purchased different items; however, since Item 2 and Item 3 both belong to Brand 1, this information implies that the two users might have a common preference for the specific brand (an implicit co-occurrence).

Since the main focus is to obtain latent representation for users and items in the recommendation task, we can explore the HIN by translating original rich semantics to user–user relationships or item–item relationships [7, 12]. In the paper, symmetric meta-paths are utilized to capture various semantic relations between two users (or items) from the HIN. The symmetric meta-path is defined as $p = nd_1 \xrightarrow{r_1} nd_2 \xrightarrow{r_2} \cdots \xrightarrow{r_{l-1}} nd_l$, where nd_1 and nd_l share the same node type. The extracted information can be transformed into user–user relationships or item–item relationships under the specific semantic, and it can be formed as a homogeneous sub-network (shown in Fig. 1b). When utilizing multiple meta-paths, we can obtain as a result a set of weighted homogeneous networks $G_h = \{G_{p_i}(V_i, E_i, w_{ij})\}$, where p_i denotes different meta-paths, (V_i, E_i) stands for the set of nodes and edges in G_{p_i} with $|o| = |r| = 1$. w_{ij} represents the weight of edge $e_{ij} \in E_i$, which is defined by the number of paths between nodes i and j.

4.2 Feature-enhanced embedding learning

We introduce an embedding learning module based on the generated homogeneous sub-networks in this section. In this paper, we aim to explore a different paradigm for building a HIN-based recommendation model. The core idea of a designed learning module is to maintain the information of heterogeneous relations via feature characterization and combine it with a neural embedding method.

4.2.1 Feature initialization strategy

To define an effective feature initialization strategy, we believe the key is to properly incorporate useful information about HIN into the model. FHetCF incorporates several encoding methods to leverage semantic and spatial correlation that are described as follows:

4.2.1.1 Semantic encoding The generated sub-networks portray the user–user (or item–item) relationships under

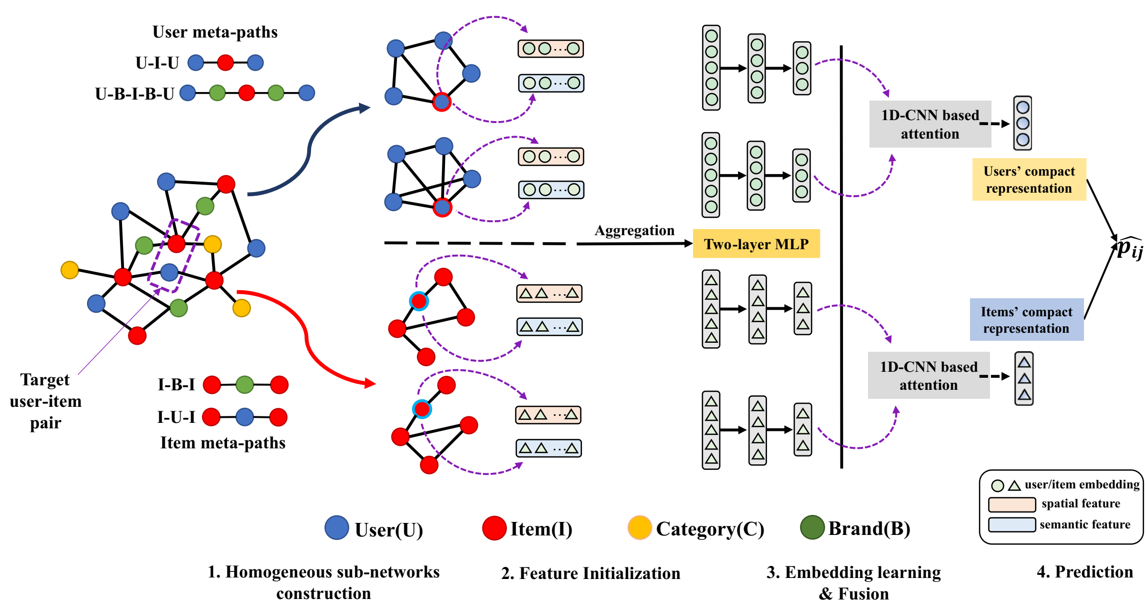


Fig. 2 The schematic demonstration of the FHetCF

specific semantics; thus, it is natural to extract semantic correlation between users (or items) for feature characterization. In FHetCF, we leverage HeteSim [60] to develop *semantic encoding*. HeteSim is a path-constrained method that calculates the similarity between two entities in heterogeneous networks. We take the symmetric odd-length relevance path to define the HeteSim measure. Given two related nodes nd_1 and nd_l, the relevance path p in the heterogeneous network is defined as $nd_1 \xrightarrow{r_1} nd_2 \xrightarrow{r_2} \cdots \xrightarrow{r_{l-1}} nd_l$, where r_i is the relation between nodes. When p is odd-length, the path can be decomposed into two equal-length paths $p_l = nd_1 nd_2 \cdots nd_{mid}$ and $p_t = nd_{mid} nd_{mid+1} \cdots nd_l$, in which nd_{mid} is the middle node of the path p. Thus, the similarity between nodes nd_1 and nd_l along the path p is defined as:

$$HeteSim(\tau_1, \tau_l|p) = HeteSim(\tau_1, \tau_l|p_l, p_t) \tag{2}$$

Furthermore, we define $X_{\tau_i \tau_j}$ is the normalized form of $A_{\tau_i \tau_j}$, which is also the transition probability matrix of $\tau_i \to \tau_j$. ($Y_{\tau_i \tau_j}$ is the transposed form of $X_{\tau_i \tau_j}$). Then, the HeteSim measure can be calculated as: (p_t is the inverse path of p_l)

$$\begin{aligned} HeteSim(\tau_1, \tau_l|p_l, p_t) &= X_{\tau_1 \tau_2} \cdots X_{\tau_{mid-1} \tau_{mid}} Y_{\tau_{mid} \tau_{mid+1}} \cdots Y_{\tau_{l-1} \tau_l} \\ &= X_{\tau_1 \tau_2} \cdots X_{\tau_{mid-1} \tau_{mid}} X^T_{\tau_{mid-1} \tau_{mid}} \cdots X^T_{\tau_1 \tau_2} \\ &= Q_{p_l} Q^T_{p_t^{-1}} \end{aligned} \tag{3}$$

As we can see, the similarity in fact can be interpreted as the inner product of two reachable probability matrices Q_{p_l} and $Q^T_{p_t^{-1}} (= Q^T_{p_l})$ that nd_1 and nd_l reaches middle node along the path p_l and p_t, respectively. According to the normalization process described in Eq. (4), we obtain the similarity vector matrix $\boldsymbol{M}^{se} = (\boldsymbol{m}_1^{se}, \boldsymbol{m}_2^{se}, \ldots, \boldsymbol{m}_n^{se})^T, \boldsymbol{M}^{se} \in \mathbb{R}^{n \times n}$ to demonstrate the semantic correlation and $m_i \in \mathbb{R}^{1 \times n}$ can be seen as the semantic feature of node i.

$$\boldsymbol{M}^{se} = \frac{Q_{p_l} Q^T_{p_t^{-1}}}{\sqrt{Q_{p_l} Q^T_{p_t^{-1}}}} \tag{4}$$

4.2.1.2 Spatial encoding An important property of network data is that nodes are distributed in a multi-dimensional spatial space and are linked by edges. We propose *spatial encoding* to capture such correlation information. For clarity, we consider a function $\varphi(v_i, v_j) : V \times V \to R$ that measures the connectivity between nodes in the sub-networks. In this paper, we choose $\varphi(v_i, v_j)$ to be the normalized symmetric weighted Laplacian matrix $\boldsymbol{M}^{sp} \in \mathbb{R}^{n \times n}$.

The Laplacian matrix is a popular matrix-form representation of a graph, and its effectiveness has been proven in capturing connectivity information [61, 62]. It calculates the gradient difference between nodes that can describe the information flow process in the connected graph from a structural perspective [63]. Thus, it is a suitable option for capturing spatial correlation in the sub-networks. For each sub-network, the weighted adjacency matrix is denoted by $\tilde{A} = (a_{ij})_{i,j=1,2,\cdots,n}$, where n is the number of nodes and a_{ij} denotes the path count. The degree d_i of node i is calculated by the equation as follows:

$$d_i = \sum_{j=1}^{n} a_{ij} \tag{5}$$

The degree matrix D is defined as the diagonal matrix with the degrees $d_1, d_2, \cdots, d_n$ on the diagonal, and I is the identity matrix. Then the spatial feature matrix M^{sp} is defined as:

$$\boldsymbol{M}^{sp} = I - D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} \tag{6}$$

4.2.1.3 Aggregation mechanism In order to characterize semantic and spatial information into node features, we use the linear combination of semantic feature $\boldsymbol{M}^{se}$ and spatial feature $\boldsymbol{M}^{sp}$ to create the initial feature $\boldsymbol{M}_0$ that served as the input for the embedding learning module:

$$\boldsymbol{M}_0 = c_1 \boldsymbol{M}^{se} + c_2 \boldsymbol{M}^{sp} \tag{7}$$

where $c_1 \geq 0$ and $c_2 \geq 0$ are the weights with $c_1 + c_2 = 1$. For simplicity, we set $c_1 = c_2 = 1/2$ in the following experiments. More complicated aggregation mechanism will be discussed in the future work. We implement the defined feature initialization process for all sub-networks.

4.2.2 Neural embedding module

With the characterized feature for each sub-network, we then learn the latent representation for different semantics. In our paper, we formulate the learning architecture via the standard MLP [64] to project nodes to low-dimensional space and preserve the information. Precisely, given the initial feature vector $\boldsymbol{M}_0^p$ generated from the semantic relation p, the projection process based on a two-layer MLP can be illustrated as follows:

$$\begin{aligned} \boldsymbol{L}_0^p &= \boldsymbol{M}_0^p \\ \boldsymbol{L}_1^p &= \sigma(\boldsymbol{W}_1 \boldsymbol{L}_0^p + \boldsymbol{b}_1) \\ \boldsymbol{E}^p &= \sigma(\boldsymbol{W}_2 \boldsymbol{L}_1^p + \boldsymbol{b}_2) \end{aligned} \tag{8}$$

where $\boldsymbol{E}^p$ stands for the obtained embedding for users or items, W_i, b_i are the weight matrix and bias for the ith hidden layer, and $\sigma(\bullet)$ is the ReLU activation function [65].

Thus, we can obtain a set of latent representations

$$E = \{E^1_{user}, E^2_{user}, \ldots, E^p_{user}, E^1_{item}, E^2_{item}, \ldots, E^p_{item}\}$$

based on all meta-paths for users and items as depicted in Fig. 2.

4.3 Meta-path level attention

As different semantics should not be equally important for the user (or item), simple fusion approaches such as averaging are not suitable for personalized recommendation with heterogeneous information. Therefore, the nonlinear weight learning method [19], which is capable of modeling complex data relations, then becomes a feasible option for fusing heterogeneous information.

Inspired by the self-attention mechanism [66], we propose a convolution-based function that learns node-wise attention [16] on meta-paths. Here we employ the one-dimensional convolutional neural network (1D-CNN) [67–70] to obtain a normalized attention score on each semantic relation. This is seen as an effective method to derive features from a fixed-length segment. Given the obtained embedding is a vector with a fixed size, we first use 1D-CNN to extract the feature as the attention weight and then perform the normalization using the softmax function [11]. Finally, the score can be used to aggregate multiple latent representations.

Specifically, given a node nd_i, we have a set of embedding vectors $e_i = \{e^1_i, e^2_i, \ldots, e^p_i\}$ obtained from P different semantic paths. 1D-CNN then maps the input into a real value g_i, which denotes the attention score on the ith meta-path:

$$g_k = \sigma\left(\sum e^k_i * W^i + b^i\right) \tag{9}$$

where e^k_i is the input embedding vector, W^i is the convolution kernel, and b^i is the corresponding offset. ReLU function is applied as the activation function in our case. $*$ represents the convolution operation, which is defined as:

$$\left(e^k_i * W^i\right)_{(m,n)} = \sum_j^{\infty} \sum_l^{\infty} \left(e^k_i\right)_{(m-j,n-l)} \left(W^i\right)_{(j,l)} \tag{10}$$

where (m, n) represents the dimension of the convolved feature of $e_i * W^i$.

Then g_i is normalized to get the final weight w_i, which measures how informative the semantic relation is to the aggregated representation of u_i:

$$\alpha_i = \frac{\exp(g_i)}{\sum_{i=1}^{p} \exp(g_i)} \tag{11}$$

With the learned personalized weight w_i, the final compact representation s_i is calculated as:

$$s_i = \sum_{k=1}^{P} \alpha_k e^k_i \tag{12}$$

4.4 Prediction and optimization

We treat the collaborative filtering as a binary classification problem that predicts the likelihood of whether a user will purchase an item. Then the preference score $\widehat{p_{ij}}$ is calculated as:

$$\widehat{p_{ij}} = \sigma\left(s_{u_i} \cdot s_{v_j}\right) \tag{13}$$

where $\sigma(\bullet)$ is the sigmoid function, and s_{u_i}, s_{v_j} are the embedding for user u_i and item v_j. In the training process, we adopt the pairwise ranking loss [] for parameters learning, which could be represented as a likelihood function:

$$p(\gamma, \gamma^- | \vartheta) = \prod_{i,j \in \gamma} \widehat{p_{ij}} \prod_{i,k \in \gamma^-} (1 - \widehat{p_{ik}}) \tag{14}$$

where γ and γ^- are the positive and negative sample set and ϑ represents the model parameters. We further employ the negative logarithm of the likelihood function as the objective function in our model:

$$L = -\sum_{i,j \in \gamma \cup \gamma^-} p_{ij} log \widehat{p_{ij}} + \left(1 - p_{ij}\right) \log\left(1 - \widehat{p_{ij}}\right) \tag{15}$$

where p_{ij} is the ground-truth label. L is then optimized by the Adam method [71] in the following experiments.

5 Experimental design

5.1 Datasets

We use two classic datasets to test the performance of our proposed model: Yelp[1] and Amazon.[2] The Yelp dataset contains users' ratings of local businesses such as hotels and restaurants collected from the Yelp challenge. The rating scores range from 1 to 5. The subset version we use contains 1,286 users and 2,614 restaurants with 30,838 reviews. The Amazon dataset covers users' rating information on products from Amazon. Our experiment chooses the electronics subset, which contains over 195,000 ratings across 6,170 users and 2,753 items.

Since we aim to investigate the recommendation performance with implicit feedback data, the explicit interactions are transformed into implicit data by introducing a binary variable, with 1 indicating that the customer has rated the item and 0 otherwise. The detailed information is presented in Table 2. (For clarity, in Yelp dataset,

[1] https://www.yelp.com/dataset challenge.

[2] http://jmcauley.ucsd.edu/data/amazon/.

Table 2 Dataset statistics

Datasets	#Entities	#Statistics
Amazon	User	6,170
	Item	2,753
	Category	22
	Brand	334
	Ratings	195,791
Yelp	User	1,286
	Item	2,614
	Category	3
	Reservation	2
	Service	2
	Star Level	9
	Ratings	30,838

'reservation' indicates whether the restaurant takes the reservation, and 'service' attribute is used to indicate whether the restaurant offers table service.)

5.2 Evaluation metrics

We adopt the leave-one-out strategy [72, 73] for evaluation. The latest interaction for each user is extracted as the test set and the remaining data is kept as the training set. For each item in the test set, we randomly sample 99 items that have no interaction records with the user as negative samples. In the evaluation procedure, we predict the probability of purchasing on the item list that containing the test item and 99 negative items. Items are ranked based on the probability and the top N will be recommended to the user.

We use hit ratio (HR) and normalized discounted cumulative gain (NDCG) as the evaluation metrics. The mathematical definitions are given in Eqs. (15) and (16):

$$HR@K = \frac{\sum Hit_i@K}{N} \tag{16}$$

$$NDCG@K = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{log_2(p_i+1)} \tag{17}$$

where $Hit_i@K$ indicates whether the ground-truth item is in the truncated list with length K for user i. NDCG@K measures the ranking accuracy of the recommendation list, N is the number of users, and p_i is the position of the ground-truth item in the user's rank list. To ensure the completeness of evaluation, we test the model performance with K = 5, 10, 15, and 20 for both metrics.

5.3 Benchmarks

To demonstrate the effectiveness of our proposed model, we compare FHetCF with some competitive benchmark models, including:

Non-neural-network methods

Itempop [51], which is a non-personalized method for recommendation. Items are simply ranked based on the number of interactions.
BPR [51], which optimizes matrix factorization with pairwise ranking loss. This is adopted for learning implicit feedback.
GMF [52], which strengthens the expressive power of the MF model by introducing the nonlinear activation function.

Neural network-based models

CMN [74], which incorporates history interaction and neighborhood information to enhance embedding learning.
NeuMF [52], which combines matrix factorization and multi-layer perception to learn latent interaction between users and items.

HIN-based method

NeuACF [11], which uses meta-paths to extract different aspects of interaction from HIN and uses multiple neural network models to learn the latent representation of users and items.
HNAFM [2], which employs a deep neural network architecture with a hierarchical attention layer to learn users' preference and items' characteristic based on meta-paths. It uses the factorization machine to perform the recommendation. To apply for top-N recommendation, we modify its optimization objective as pairwise ranking loss as in BPR.
LGRec [20], which is designed to exploit different interactions information embedded in the HIN. The latent vectors for users and items are obtained through the combination of representations of direct interacted neighbors and complex meta-path interactions.

5.4 Experimental settings

Meta-paths used in this study to construct homogeneous sub-networks are shown in Table 3. The hyper-parameters setting of our proposed method stays the same for the two datasets. We tune the hyperparameters using a grid search, where the batch size and learning rate were searched in [128, 256, 512, 1024] and [1e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-

Table 3 Meta-paths used in each dataset

Dataset	Meta-paths
Amazon	UIU, UIBIU, UICIU
	IUI, IBI, ICI
Yelp	UBU, UBReBU, UBSeBU, UBStBU, UBCaBU
	BUB, BReB, BSeB, BStB, BCaB

2], respectively. Here, we adopt a simple two-layer MLP for embedding learning with the number of hidden units in the first and the second layer set to 512 and 64, respectively. Besides, a two-layer one-dimensional convolution neural network (1D-CNN) model is used to learn the personalized preference on meta-paths. Specifically, we set kernel size and stride to be 1, and the number of filters in the first and the second layer to 16 and 1, respectively. As for the negative sample set in training process, we empirically set the negative sampling rate to 10. We keep the parameter setting of benchmark models the same as the one stated in their original paper in our experiment. All experiments are implemented in Python with Tensorflow [75].

6 Results

6.1 Overall performance comparison

To ensure a fair comparison with the benchmark models, we test the model performance on different recommendation datasets. For FHetCF and baselines, we report the average result of ten time runs on the two datasets in Tables 4 and 5, respectively.

In terms of HR and NDCG, we can see that our FHetCF achieves the best recommendation performance on both datasets and criteria. FHetCF significantly outperforms the second-best performing method by a large margin. Specifically, for the Amazon dataset, the relative improvement over the strongest baseline NeuACF is 2.2% and 3.57% on average for HR and NDCG, respectively. For the Yelp dataset, there is an increase of 7.1% in hit ratio and 11.68% in NDCG on average against second best performing baselines. These findings verify the effectiveness of our proposed model. The performance improvement may come from two aspects: on the one hand, the embeddings generated from the high-quality initial feature can encode important properties of different types of nodes; on the other, 1D-CNN-based attention mechanism can better determine the importance of different semantic relations.

The performance of Itempop is rather poor on both datasets and largely falls behind the other benchmark models. Note that Itempop is a non-personalized method

Table 4 Performance comparison on Yelp dataset*

	HR@5	NDCG@5	HR@10	NDCG@10	HR@15	NDCG@15	HR@20	NDCG@20
Itempop	0.2932	0.2011	0.4448	0.2498	0.5365	0.2742	0.6065	0.2907
BPR	0.5309	0.3712	0.6943	0.4222	0.7853	0.4470	0.8404	0.4599
GMF	0.5447	0.3798	0.7135	*0.4447*	0.7827	0.4562	0.8469	*0.4809*
CMN	0.5427	0.3809	0.7004	0.4321	0.7863	0.4549	0.8405	0.4677
NeuMF	0.5500	0.3854	0.7199	0.4429	0.7971	0.4611	0.8605	*0.4809*
HNAFM	*0.5591*	0.3844	0.7216	0.4373	0.7893	0.4553	0.8227	0.4631
LGRec	0.5580	0.3863	*0.7291*	0.4420	*0.8201*	*0.4661*	*0.8763*	0.4794
NeuACF	0.5465	*0.3882*	0.7124	0.4419	0.8039	*0.4661*	0.8611	0.4797
FHetCF	**0.6176**	**0.4416**	**0.7893**	**0.4976**	**0.8659**	**0.5178**	**0.9121**	**0.5288**
Improved	+ 10.46%	+ 13.75%	+ 8.26%	+ 11.9%	+ 5.58%	+ 11.10%	+ 4.09%	+ 9.96%

*The best and second-best scores are marked in bold and italics, respectively

Table 5 Performance comparison on Amazon dataset*

	HR@5	NDCG@5	HR@10	NDCG@10	HR@15	NDCG@15	HR@20	NDCG@20
Itempop	0.1828	0.1167	0.2831	0.1489	0.3684	0.1715	0.4365	0.1875
BPR	0.2847	0.1863	0.4212	0.2150	0.5237	0.2569	0.6022	0.2775
GMF	0.2632	0.1755	0.3929	0.2169	0.4936	0.2430	0.5642	0.2619
CMN	0.2702	0.1777	0.4114	0.2231	0.5123	0.2498	0.5874	0.2675
NeuMF	0.2884	0.1924	0.4273	0.2331	0.5331	0.2619	0.6095	0.2802
HNAFM	0.2248	0.1458	0.3653	0.1910	0.4600	0.2161	0.5321	0.2331
LGRec	0.2737	0.1819	0.4142	0.2271	0.5141	0.2535	0.5902	0.2715
NeuACF	*0.3147*	*0.2097*	*0.4636*	*0.2576*	*0.5677*	*0.2851*	*0.6429*	*0.3029*
FHetCF	**0.3237**	**0.2185**	**0.4747**	**0.2671**	**0.5774**	**0.2943**	**0.6546**	**0.3125**
Improved	+ 2.86%	+ 4.2%	+ 2.39%	+ 3.69%	+ 1.71%	+ 3.23%	+ 1.82%	+ 3.17%

*The best and second-best scores are marked in bold and italics, respectively that simply recommends the items based on interaction statistics; personalized preference is completely ignored in the interaction modeling process. It also can be observed that GMF and BPR yield much better performance than Itempop. GMF and BPR are both matrix factorization-based models that can generate a latent feature for each user or item that describes preference or characteristics to facilitate the interaction modeling, thus improving the recommendation performance.

NeuMF and CMN generally achieve better performance than the non-neural network models on both the datasets. The result admits that the neural network has superior ability in learning latent factors when compared to linear models. Moreover, NeuMF surpasses the performance of CMN all the time and perform well on most conditions, indicating that a combination of the MF and MLP models fuses linear and nonlinear information and is an effective approach to modeling complex user–item interaction.

As can be seen from the experimental results, the HIN-based methods substantially obtain the superior performance in most cases among all kinds of baselines. On the Amazon dataset, the performance gains of NeuACF are on average 7.39% and 9.12% for HR and NDCG, respectively (when compared to NeuMF). For the Yelp dataset, LGRec generally outperforms other models. An intuitive explanation is that HIN-based methods could better capture diverse auxilliary information and higher-order interaction patterns hidden in the HINs. Notably, the proposed FHetCF based on meta-paths is able to effectively integrate rich semantic and spatial information into interaction modeling.

6.2 Ablation study

6.2.1 Comparison on different initialization strategies

In this section, we implement a comparative analysis of existing feature initialization strategies based on network information in HIN-related research and investigate their performance in recommendation. Common initialization choices in existing research roughly fall into two classes:

Random initialization: We discuss two commonly used distributions in the experiment and develop corresponding variants of FHetCF, namely FHetCF-xav for Xavier initialization and FHetCF-uni for standard Gaussian initialization. The dimension of random feature vector is set to be 512.

Meta-path-based feature: We develop variants of FHetCF using a meta-path-based feature. FHetCF-pc uses normalized commuting matrix (definition given in Sect. 3) as the input feature. FHetCF-one is the unweighted version of FHetCF-pc, with $c_{ij} > 0$ transformed to $c_{ij} = 1$ in the commuting matrix C.

To ensure a fair comparison under the unified setting, we exclude strategies such as bag of words [43] and pre-train techniques [18] in our analysis since these introduce additional information to obtain the initial feature. Performance is compared based on the average score of ten times run. The experimental results on the two datasets are presented in Tables 6 and 7.

Table 6 Performance of FHetCF on Yelp with different feature initialization strategies*

	HR@5	NDCG@5	HR@10	NDCG@10	HR@15	NDCG@15	HR@20	NDCG@20
FHetCF-uni	0.3159	0.2160	0.4536	0.2604	0.5447	0.2845	0.6107	0.3001
FHetCF-xav	0.4474	0.3058	0.6224	0.3625	0.7161	0.3873	0.7875	0.4042
FHetCF-pc	0.3926	0.2711	0.5672	0.3274	0.6572	0.3512	0.7286	0.3681
FHetCF-one	*0.5781*	*0.4056*	*0.7451*	*0.4598*	*0.8218*	*0.4801*	*0.8732*	*0.4922*
FHetCF	**0.6176**	**0.4416**	**0.7893**	**0.4976**	**0.8659**	**0.5178**	**0.9121**	**0.5288**
Improved	+ 6.832%	+ 8.876%	+ 5.932%	+ 8.221%	+ 5.366%	+ 7.853%	+ 4.455%	+ 7.436%

*The best and second−best scores are marked in bold and italics, respectively

Table 7 Performance of FHetCF on Amazon with different feature initialization strategies*

	HR@5	NDCG@5	HR@10	NDCG@10	HR@15	NDCG@15	HR@20	NDCG@20
FHetCF-uni	0.1505	0.0956	0.2529	0.1284	0.3353	0.1502	0.4069	0.1671
FHetCF-xav	0.2209	0.1428	0.3487	0.1840	0.4444	0.2092	0.5222	0.2276
FHetCF-pc	0.1957	0.1273	0.3145	0.1655	0.3999	0.1880	0.4743	0.2056
FHetCF-one	*0.2991*	*0.1977*	*0.4496*	*0.2462*	*0.5535*	*0.2737*	*0.6316*	*0.2921*
FHetCF	**0.3237**	**0.2185**	**0.4747**	**0.2671**	**0.5774**	**0.2943**	**0.6546**	**0.3125**
Improved	+ 8.225%	+ 10.52%	+ 5.583%	+ 8.489%	+ 4.318%	+ 7.526%	+ 3.642%	+ 6.984%

*The best and second−best scores are marked in bold and italics, respectively

With regard to random initialization, we can see that Xavier initialization obtains a much better performance than the standard Gaussian initialization on both datasets. As stated in [76], after the standard Gaussian initialization, the variance of the back-propagated gradients gets smaller layer by layer, leading to the vanishing gradient problem during learning. Xavier initialization keeps the same variance of activations across every layer to prevent gradients from vanishing. The results of our experiment demonstrate the effectiveness of Xavier initialization in embedding learning task.

As for the meta-path-based feature, we can find that FHetCF-pc largely falls behind FHetCF-one based on HR@K and NDCG@K metrics. It is also worth noting that random initialized model FHetCF-xav generally outperforms FHetCF-pc. The comparison shows that a normalized commuting matrix might not be a proper way to encode the feature. The variant FHetCF-one beats all other variants on the Yelp and Amazon datasets, indicating the importance of a proper method for characterizing the feature.

It can be observed that the proposed FHetCF method yields consistently better recommendation performance than FHetCF-one. The improvements of FHetCF over FHetCF-one on the Yelp dataset with regard to HR@K are 6.6, 5.28, 5.18, and 4.55% when K equals 5, 10, 15, and 20, respectively. Moreover, FHetCF shows a 6.8 to 8% improvement over the variant on the NDCG@K metric with different K, demonstrating the generated item list can approximately reflect a user's preference.

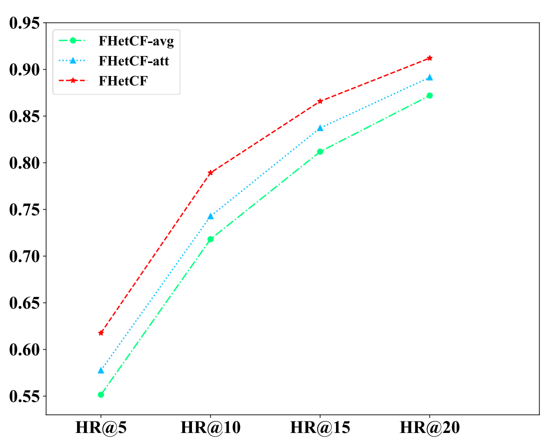
On the Amazon dataset, the relative improvements over the FHetCF-one with regard to NDCG@K are 10.37, 8.245, 7.344, and 6.607% (K = 5, 10, 15, and 20, respectively). For the HR@K metric, FHetCF gains a 3.3 to 8.49% improvement over the variant with varying K.

The overall results validate the effectiveness of the proposed feature-enhanced learning architecture and indicate that encoding the semantic similarity and spatial correlation into the node feature can improve recommendation performance in comparison with other feature initialization strategies.

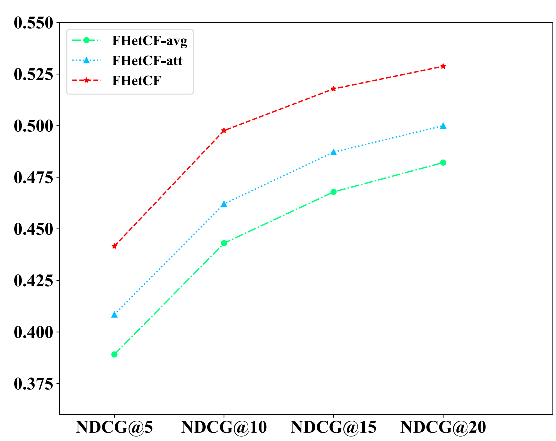
6.2.2 Effect of attention mechanism

To investigate the effect of the proposed fusion mechanism, we implement experiments to compare it with two commonly used fusion functions. We employ 1D-CNN to distinguish the importance of different meta-paths (semantics) and model the users' preference into embedding fusion. In particular, equal weight and self-attention mechanisms are chosen to be variant models for comparison, named FHetCF-avg and FHetCF-att, respectively. The comparison results are shown in Figs. 3 and 4, respectively.

FHetCF-avg performs the worst of the three models according to evaluation metrics. The reason for this is that it treats each type of relation with equal importance and neglects personalized preference information when modeling the user–item interaction. We observe from Figs. 3 and 4 that FHetCF-att with attention mechanism obtains a much greater recommendation performance than FHetCF-avg. For the Yelp dataset, FHetCF-att improves over the FHetCF-avg by 2.24 to 4.73% and 3.7 to 4.96% on HR and NDCG, respectively. On the Amazon dataset, the performance gains are observed with a minimum improvement of 2.55 and 3.64% on HR and NDCG, respectively. The comparison indicates the necessity of incorporating personalized weight into embedding fusion. Our proposed method acquires the best recommendation performance on HR@K and NDCG@K.



(a) Hit Ratio



(b) NDCG

Fig. 3 Performance of FHetCF on Yelp with different fusion mechanisms

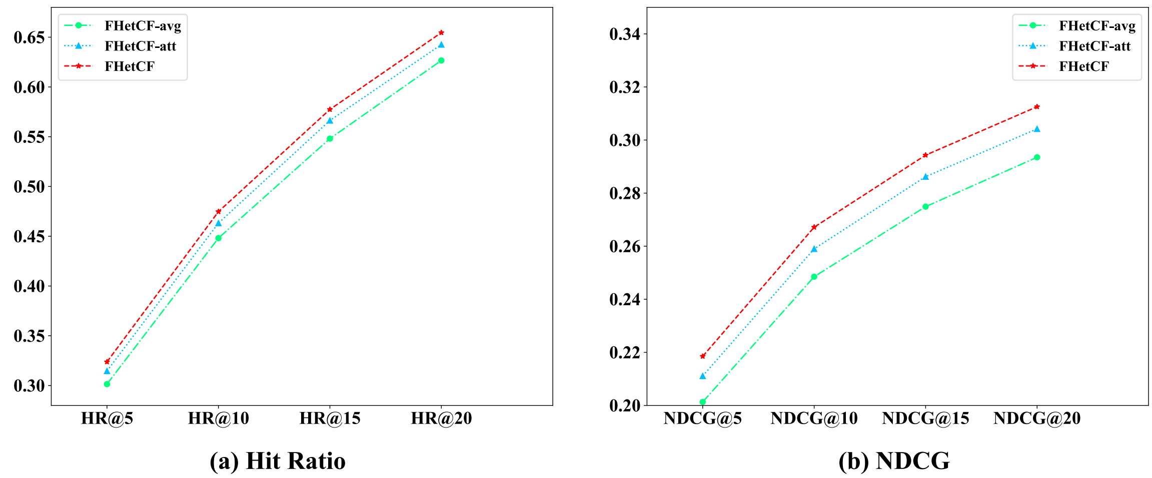


Fig. 4 Performance of FHetCF on Amazon with different fusion mechanisms

Considering the Yelp dataset, introducing 1D-CNN leads to 6.17% and 10.71% improvements on average over the attention mechanism and equal weight on NDCG@K. For hit ratio, the improvements of FHetCF over FHetCF-avg with regard to HR@K are 11.76, 9.24, 6.46, and 4.69% when K equals to 5, 10, 15, and 20, respectively. The improvements of FHetCF over FHetCF-att with regard to HR@K are 6.7, 5.61, 3.24, and 2.41%, respectively.

It can also be observed that on the Amazon dataset, our proposed fusion mechanism improves by 2.3 and 5.8% over the attention mechanism and equal weight on HR@K. For NDCG, the improvements of FHetCF over FHetCF-avg with regard to NDCG@K are 8.4, 7.24, 6.87, and 6.1% when K equals 5, 10, 15, and 20, respectively. The improvements of FHetCF over FHetCF-att with regard to NDCG@K are 3.36, 2.89, 2.65, and 2.37%, respectively.

The experimental results validate the rationality of our proposed fusion mechanism and demonstrate the effectiveness of applying 1D-CNN to automatically determine importance.

6.2.3 Impact of meta-paths

To analyze the impact of meta-paths on the model performance, we compare the performance of FHetCF with and without meta-paths and run the FHetCF with single meta-path. In Fig. 5, the results show the benefit of heterogeneous information embodied in the meta-path. By comparing the performance on Amazon and Yelp, we find that more useful heterogeneous information (more meta-paths) can better enhance the recommendation performance.

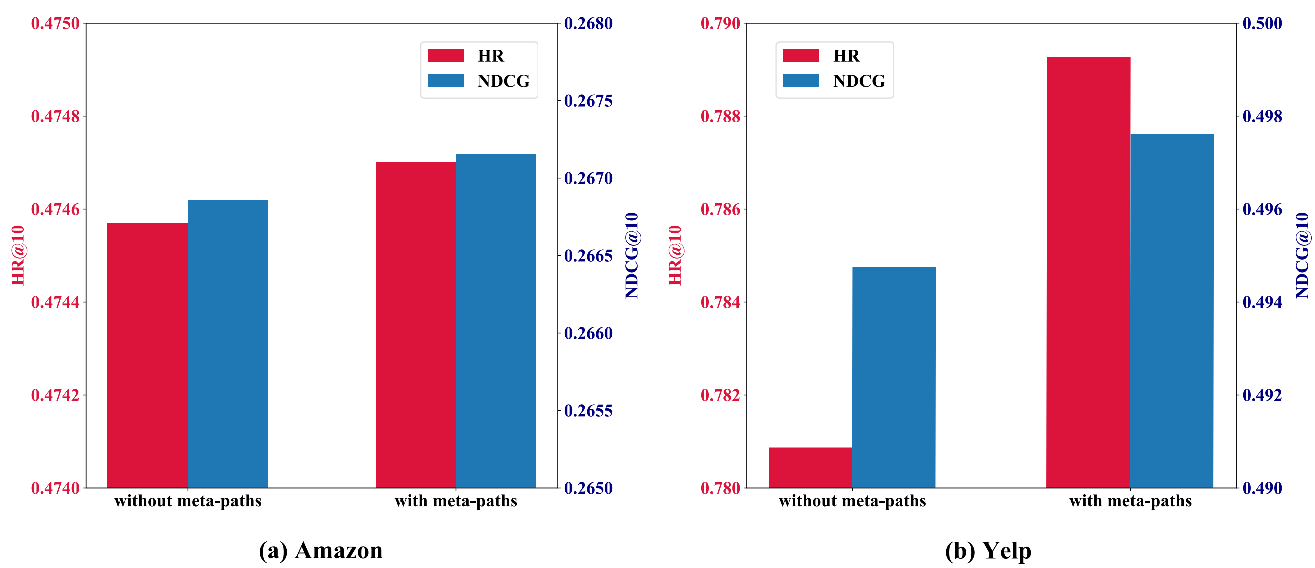


Fig. 5 The impact of introducing various meta-paths

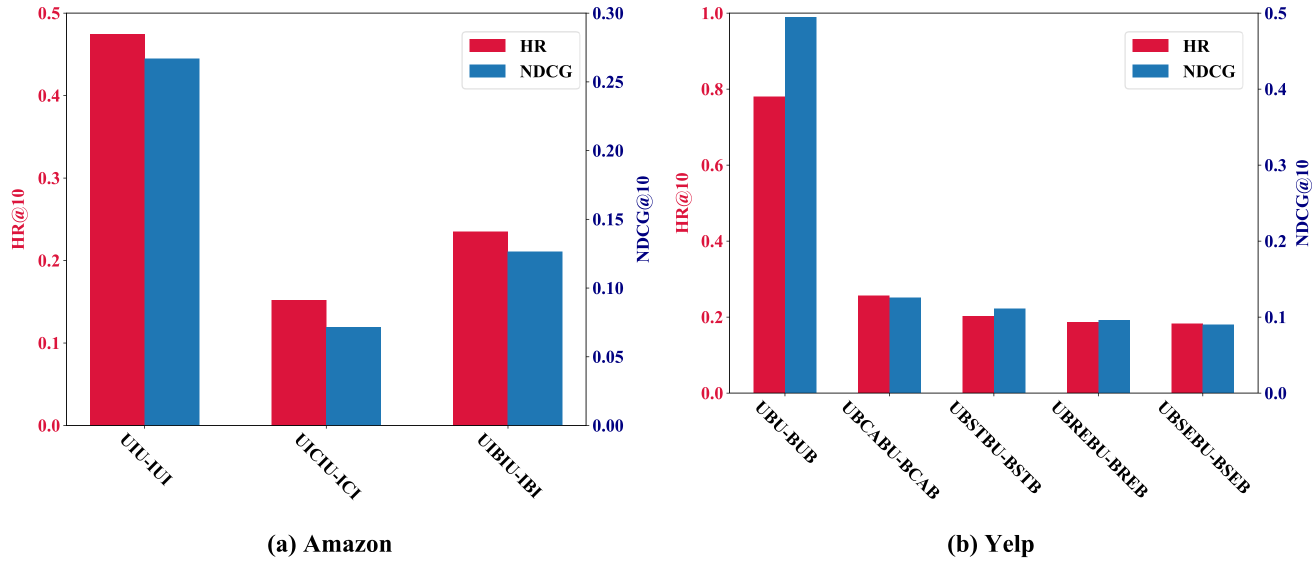


Fig. 6 The impact of different single meta-paths

In Fig. 6, for instance, *UCICU-ICI* refers to we only utilize the user preference and item characteristic information obtained from the *UCICU-ICI* meta-path to make recommendation. It can be seen from Fig. 6, *UIU-IUI/UBU-BUB* meta-path gets the best performance among all the meta-paths since it conveys the most important information for recommendation which indicates the user-item historical interaction pattern. Meanwhile, we can also observe that other meta-paths contain a certain amount of additional information which is capable of improving the recommendation performance (as validated in Sect. 6.1).

7 Conclusions

In this paper, we introduce a feature-enhanced embedding learning model, denoted as FHetCF, for heterogeneous collaborative filtering. The general idea is to combine an informative feature initialization strategy with a relatively simple learning architecture. We first use symmetric meta-paths to extract various semantic relations hidden in the HIN and construct corresponding homogenous sub-networks. We then design an initialization strategy for which *semantic encoding* and *spatial encoding* are implemented to characterize the node feature. Further, we design a node-wise heterogeneous information fusion algorithm that learns the importance of different semantic relations (meta-paths) for users and items, respectively. The comparison experiments and extensive ablation studies show the superiority of our proposed model over competitive recommendation methods. Moreover, the results validate that the proposed architecture design is capable of acquiring promising recommendation performance.

It is worth noting that we restrict our discussion in this study to a situation whereby network information is the only available feature, and the purpose is to discuss how to fully exploit available information for better recommendations. Meanwhile, the proposed model can be seen as a general framework for heterogeneous collaborative filtering, and we can improve the recommendation performance via a more comprehensive initialization strategy. For example, future work can explore different feature encoding methods, such as shortest path calculation for spatial encoding, or more information can be added to the strategy, such as node centrality and eigenvector centrality. We can also introduce methods such as bilinear interaction for feature aggregation.

Acknowledgements X.L. and Y.T. acknowledges the Natural Science Foundation of China under Grant 72025405, 91846301, 72088101and 71790615, and the Innovation Team Project of Colleges in Guangdong Province (2020KCXTD040). W.Y. are partially supported by the Natural Science Foundation of China under Grant 71690233. J.L. are partially supported by the Natural Science Foundation of China under Grant 71690233 and 72001209. S.T. are partially supported by the National Natural Science Foundation of China (72001211) and the Hunan Science and Technology Plan Project (2020JJ5679 and 2020TP1013).

Declarations

Conflict of interest We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Fan S, Zhu J, Han X, et al (2019) Metapath-guided heterogeneous graph neural network for intent recommendation[C]. In:

Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp 2478–2486
2. Chen L, Liu Y, Zheng Z, et al (2018) Heterogeneous neural attentive factorization machine for rating prediction[C]. In: Proceedings of the 27th ACM international conference on information and knowledge management. pp 833–842
3. Yang X, Li Y, Wang H, et al (2019) Bid optimization by multivariable control in display advertising[C]. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp 1966–1974
4. Simsek A, Karagoz P (2020) Wikipedia enriched advertisement recommendation for microblogs by using sentiment enhanced user profiles[J]. J Intell Inform Syst 54(2):245–269
5. Lian J, Zhou X, Zhang F, et al (2018) xdeepfm: Combining explicit and implicit feature interactions for recommender systems[C]. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp 1754–1763
6. Pinto FM, Estefania M, Cerón N, et al (2016) iRecomendYou: a design proposal for the development of a pervasive recommendation system based on student's profile for Ecuador's students' candidature to a scholarship[M]. In: New advances in information systems and technologies. Springer, Cham, pp 537–546
7. Zhang Y, Xiong Y, Kong X, et al (2018) Deep collective classification in heterogeneous information networks[C]. In: Proceedings of the 2018 world wide web conference. pp 399–408
8. Goldberg D, Nichols D, Oki BM, et al (1992) Using collaborative filtering to weave an information tapestry[J]. Commun ACM 35(12):61–70
9. Su X, Khoshgoftaar T M (2009) A survey of collaborative filtering techniques [J]. Adv Artif Intell 2009(12)
10. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems[J]. Computer 42(8):30–37
11. Han X, Shi C, Wang S, et al (2018) Aspect-Level Deep Collaborative Filtering via Heterogeneous Information Networks[C]. In: Proceedings of the 27th international joint conference on artificial intelligence, vol 18. pp 3393–3399
12. Zhao J, Zhou Z, Guan Z, et al (2019) Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation[C]. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp 2347–2357
13. Eswaran D, Günnemann S, Faloutsos C, et al (2017) Zoobp: Belief propagation for heterogeneous networks[J]. Proc VLDB Endowment 10(5):625–636
14. Yang C, Liu M, He F, et al (2018) Similarity modeling on heterogeneous networks via automatic path discovery[C]. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, Cham, pp 37–54
15. Zhang W, Paudel B, Wang L, et al (2019) Iteratively learning embeddings and rules for knowledge graph reasoning[C]. In: The world wide web conference. pp 2366–2377
16. Han Z, Anwaar M U, Arumugaswamy S, et al (2020) Metapath- and entity-aware graph neural network for recommendation[J]. CoRR, abs/2010.11793
17. Wang J, Huang P, Zhao H, et al (2018) Billion-scale commodity embedding for e-commerce recommendation in alibaba[C]. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp 839–848
18. Hu B, Shi C, Zhao W X, et al (2018) Leveraging meta-path based context for top-n recommendation with a neural co-attention model[C]. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp 1531–1540
19. Shi C, Hu B, Zhao WX, et al (2018) Heterogeneous information network embedding for recommendation[J]. IEEE Trans Knowl Data Eng 31(2):357–370
20. Hu B, Shi C, Zhao W X, et al (2018) Local and global information fusion for top-n recommendation in heterogeneous information network[C]. In: Proceedings of the 27th ACM international conference on information and knowledge management. pp 1683–1686
21. Zhao Z, Zhang X, Zhou H, et al (2020) HetNERec: heterogeneous network embedding based recommendation[J]. Knowl-Based Syst 204:106218
22. Jin J, Du K, Zhang W, et al (2020) GraphHINGE: Learning interaction models of structured neighborhood on heterogeneous information network[J]. arXiv preprint arXiv:2011.12683
23. Zhao H, Yao Q, Song Y, et al (2021) Side information fusion for recommender systems over heterogeneous information network[J]. ACM Trans Knowl Discov Data 15(4):1–32
24. Xie F, Zheng A, Chen L, et al (2021) Attentive meta-graph embedding for item recommendation in heterogeneous information networks[J]. Knowl-Based Syst 211:106524
25. Pham TAN, Li X, Cong G, et al (2016) A general recommendation model for heterogeneous networks[J]. IEEE Trans Knowl Data Eng 28(12):3140–3153
26. Liang H (2020) Drprofiling: deep reinforcement user profiling for recommendations in heterogenous information networks[J]. IEEE Trans Knowl Data Eng 2020
27. Xu Y, Chen H, Deng Z, et al (2020) Single-layer graph convolutional networks for recommendation[J]. arXiv preprint arXiv:2006.04164
28. Wang Z, Liu H, Du Y, et al (2019) Unified embedding model over heterogeneous information network for personalized recommendation[C]. In: Proceedings of the 28th international joint conference on artificial intelligence, vol 2019. pp 3813–3819
29. Shi J, Ji H, Shi C, et al (2020) Heterogeneous graph neural network for recommendation[J]. arXiv preprint arXiv:2009.00799
30. Sun Y, Han J, Yan X, et al (2011) Pathsim: meta path-based top-k similarity search in heterogeneous information networks[J]. Proc VLDB Endowment 4(11):992–1003
31. LeCun Y, Bottou L, Bengio Y, et al (1998) Gradient-based learning applied to document recognition[J]. Proc IEEE 86(11):2278–2324
32. Hamilton W L, Ying R, Leskovec J (2017) Representation learning on graphs: methods and applications[J]. arXiv preprint arXiv:1709.05584
33. Santos LD, Piwowarski B, Gallinari P (2016) Multilabel classification on heterogeneous graphs with gaussian embeddings[C]. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, Cham, pp 606–622
34. Chen T, Sun Y (2017) Task-guided and path-augmented heterogeneous network embedding for author identification[C]. In: Proceedings of the 10th ACM international conference on web search and data mining. 2017: 295–304.
35. Tommasel A, Godoy D (2016) Integrating heterogeneous information from social networks into community detection[C]. In: 4th IJCAI workshop on heterogeneous information network analysis (HINA)
36. Luo C, Pang W, Wang Z (2014) Semi-supervised clustering on heterogeneous information networks[C]. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, Cham, pp 548–559
37. Yang C, Xiao Y, Zhang Y, et al (2020) Heterogeneous network representation learning: a unified framework with survey and benchmark[J]. IEEE Trans Knowl Data Eng 2020
38. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations[C]. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. pp 701–710
39. Dong Y, Chawla N V, Swami A (2017) metapath2vec: Scalable representation learning for heterogeneous networks[C]. In:

Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp 135–144
40. Fu T, Lee W C, Lei Z (2017) Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning[C]. In: Proceedings of the 2017 ACM on conference on information and knowledge management. pp 1797–1806
41. Tang J, Qu M, Mei Q (2015) Pte: Predictive text embedding through large-scale heterogeneous text networks[C]. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp 1165–1174
42. Schlichtkrull M, Kipf TN, Bloem P, et al (2018) Modeling relational data with graph convolutional networks[C]. In: European semantic web conference. Springer, Cham, pp 593–607
43. Wang X, Ji H, Shi C, et al (2019) Heterogeneous graph attention network[C]. In: The world wide web conference. pp 2022–2032
44. Fu X, Zhang J, Meng Z, et al (2020) MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding[C]. In: Proceedings of the web conference, vol 2020. pp 2331–2341
45. Hu Z, Dong Y, Wang K, et al (2020) Heterogeneous graph transformer[C]. In: Proceedings of the web conference, vol 2020. pp 2704–2710
46. Hong H, Guo H, Lin Y, et al (2020) An attention-based graph neural network for heterogeneous structural learning[C]. Proc AAAI Conf Artif Intell 34(04):4132–4139
47. Yu L, Sun L, Du B, et al (2021) Heterogeneous graph representation learning with relation awareness[J]. arXiv preprint arXiv:2105.11122
48. Zhao J, Wang X, Shi C, et al (2021) Heterogeneous graph structure learning for graph neural networks[C]. Proc AAAI Conf Artif Intell 35(5):4697–4705
49. Liu Z, Zheng V W, Zhao Z, et al (2018) Interactive paths embedding for semantic proximity search on heterogeneous graphs[C]. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp 1860–1869
50. Qu Y, Bai T, Zhang W, et al (2019) An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation[C]. In: Proceedings of the 1st international workshop on deep learning practice for high-dimensional sparse data, vol 2019, pp 1–9
51. Rendle S, Freudenthaler C, Gantner Z, et al (2012) BPR: Bayesian personalized ranking from implicit feedback[J]. arXiv preprint arXiv:1205.2618
52. He X, Liao L, Zhang H, et al (2017) Neural collaborative filtering[C]. In: Proceedings of the 26th international conference on world wide web. pp 173–182
53. Wang X, He X, Wang M, et al (2019) Neural graph collaborative filtering[C]. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp 165–174
54. Yu X, Chu Y, Jiang F, et al (2018) SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features[J]. Knowl-Based Syst 141:80–91
55. Cantador I, Fernández-Tobías I, Berkovsky S, et al (2015) Cross-domain recommender systems[M]. In: Recommender systems handbook. Springer, Boston, pp 919–959
56. Yu X, Jiang F, Du J, et al (2019) A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains[J]. Pattern Recogn 94:96–109
57. Yu X, Peng Q, Xu L, et al (2021) A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm[J]. Inf Process Manage 58(6):102691
58. Qian C, Yu Y, Zhou Z H (2015) Pareto ensemble pruning[C]. In: 29th AAAI conference on artificial intelligence
59. Yang Y, Guan Z, Li J, et al (2021) Interpretable and efficient heterogeneous graph convolutional network [J]. IEEE Trans Knowl Data Eng
60. Shi C, Kong X, Huang Y, et al (2014) Hetesim: a general framework for relevance measure in heterogeneous networks[J]. IEEE Trans Knowl Data Eng 26(10):2479–2492
61. Kipf T N, Welling M (2016) Semi-supervised classification with graph convolutional networks [J]. arXiv preprint arXiv:1609.02907
62. Von Luxburg U (2007) A tutorial on spectral clustering[J]. Stat Comput 17(4):395–416
63. Chung F R K, Graham F C (1997) Spectral graph theory[M]. Am Math Soc
64. Pal SK, Mitra S (1992) Multilayer perceptron, fuzzy sets, classification [J]. IEEE Trans Neural Networks 3(5):683
65. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks[C]. In: Proceedings of the 14th international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings. pp 315–323
66. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need [J]. Adv Neural Inf Process Syst 30
67. Kalchbrenner N, Grefenstette E, Blunsom P, et al (2020) A Convolutional Neural Network for Modelling Sentences[C]. In: Proceedings of the 52nd annual meeting of the association for computational linguistics. Association for computational linguistics. pp 212–217
68. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions [J]. arXiv preprint arXiv:1511.07122
69. Wang Y, Tian L, Li C (2020) An improved text classification model based on memory convolution neural network[C]. In: Proceedings of the 2020 6th international conference on computing and artificial intelligence. pp 19–23
70. Gao M, Dong J, Zhou D, et al (2019) End-to-end speech emotion recognition based on one-dimensional convolutional neural network[C]. In: Proceedings of the 2019 3rd international conference on innovation in artificial intelligence. pp 78–82
71. Kingma D P, Ba J (2014) Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980
72. Bayer I, He X, Kanagal B, et al (2017) A generic coordinate descent framework for learning from implicit feedback[C]. In: Proceedings of the 26th international conference on world wide web. pp 1341–1350
73. He X, Zhang H, Kan M Y, et al (2016) Fast matrix factorization for online recommendation with implicit feedback[C]. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval. pp 549–558
74. Ebesu T, Shen B, Fang Y (2018) Collaborative memory network for recommendation systems[C]. In: The 41th international ACM SIGIR conference on research & development in information retrieval. pp 515–524
75. Abadi M (2016) TensorFlow: learning functions at scale[C]. In: Proceedings of the 21th ACM SIGPLAN international conference on functional programming. p 1
76. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks[C]. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings. pp 249–256