Semantic enhanced Top-k similarity search on weighted HIN

Yun Zhang, Minghe Yu, Tiancheng Zhang, Ge Yu

Received: 21 May 2021 / Accepted: 19 April 2022 / Published online: 18 June 2022
© The Author(s) 2022

Abstract
Similarity searches on heterogeneous information networks (HINs) have attracted wide attention from both industrial and academic areas in recent years; for example, they have been used for friend detection in social networks and collaborator recommendation in coauthor networks. The structural information on the HIN can be captured by multiple metapaths, and people usually utilize metapaths to design methods for similarity search. The rich semantics in HINs are not only structural information but also content stored in nodes. However, the content similarity of nodes was usually not valued in the existing methods. Although some researchers have recently considered both types of information in machine learning-based methods for similarity search, they have used structure and content information separately. To address this issue by balancing the influence of structure and content information flexibly in the process of searching, we propose a double channel convolutional neural network model for top-k similarity search, which uses path instances as model inputs and generates structure and content embeddings for nodes based on different metapaths. We design an attention mechanism to enhance the differences in metapaths for each node. Another attention mechanism is used to combine the content and structure information of nodes. Finally, an importance evaluation function is designed to improve the accuracy and make the model more explainable. The experimental results show that our search algorithm can effectively support top-k similarity search in HINs and achieve higher performance than existing approaches.

Keywords Attention mechanism · Convolutional neural network · Weighted heterogeneous information network · Representation learning · Top-k similarity search

1 Introduction

In recent years, data mining on heterogeneous information networks has attracted extensive attention from both industrial and academic areas. For example, in an academic social network, users can search the most similar author or the most relevant paper based on the queries they input [1]. Similarity search is a popular data mining task based on strategies to find the objects most similar to a given object (query statement, node, etc.). Classical similarity search studies, including PageRank [2], HITS [3], and SimRank [4], focus on homogeneous information networks or binary networks, which have only one type of edge. However, data in the real world are very complicated; thus using a homogeneous information network to describe data cannot completely express the rich semantic information. Therefore, a heterogeneous information network (HIN) has become a great option to describe real data. Recently, searching and recommendation technologies in HINs have been gradually proposed [5–15], which aim to find similar objects in HINs based on given information.

Specifically, the main goal of the top-k similarity search task in HINs we will study is to obtain a set of nodes related to a given node according to the network characteristics. At present, predefined metapaths that can capture structural information are widely used in HIN-based research. Unlike in homogeneous information networks, different metapaths in HINs are specified to represent different semantics. Therefore, it is better to flexibly adjust different metapaths for different queries. Since nodes have both structural information and content information, how to
integrate them to comprehensively represent nodes is also a great challenge that needs to be solved.

There are some unsupervised methods [6, 16] that consider the content and structure information of nodes for similarity search in HINs. However, these methods utilize a fixed mode to combine them. In fact, for nodes with few interactions, capturing their content information is more important. Meanwhile, for active nodes, the main task is to train their structure information in the given HIN for similarity search. Therefore, it is necessary to flexibly consider the content information and structure information of nodes.

Machine learning-based methods utilize node embedding in HINs for similarity search, that is, measure the similarity of nodes according to their vectors’ similarity. Metapath2vec [17] and some other methods [5, 18–24] only represent nodes based on one metapath, which is useful for tasks with specific semantic requirements. For recommendation tasks, MCRRec [11] uses an attention mechanism to combine multiple metapath information to realize similarity measurement between users and items, but it does not consider the content information of nodes. HetGNN [16], which considers the content and structure information on nodes, uses the attention mechanism for embedding, while the content and structure information are not comprehensively combined. Specifically, when the model is trained by the nodes’ structure information, the content features fade or even disappear. In addition, none of the above methods considers the time factor contained in the network. Therefore, it is possible that two nodes with completely opposite changing trends are considered to be the most similar. In addition, none of the above methods consider the time factor contained in the network. Therefore, it is possible to consider two nodes with completely opposite trends as the most similar. In addition, some data distribution information in the network will also affect the characteristics of nodes.

To achieve the similarity search task in HIN and to overcome the above challenges, including adjusting different metapaths for different queries, flexibly integrating structure information and content information to comprehensively represent nodes, and considering the time factor and other data distribution information, we propose a model for top-k similarity search based on a double channel convolutional neural network in weighted HINs (called SSDCC+), which uses a CNN with two channels to train the content information and structure information at the same time. We generate different structure and content embeddings for each node according to different metapaths, which has not been done in other previous studies, and extract the internal attribute information of the node. We sample path instances for given metapaths and calculate their weights. To train content information and structure information at the same time, a double channel CNN is used in SSDCC+. Our model can distinguish the metapaths for each node by an attention mechanism because the importance of metapaths is not the same for each node. Another attention mechanism, which considers the outputs of the two channels and the attribute information of the data, is proposed to combine the node’s content and structure information for its comprehensive representation. Finally, an evaluation function is designed using the time information contained in the objects and the data distribution information to evaluate the importance of the content and structure information.

To summarize, the major contributions of our work are as follows:

- To the best of our knowledge, this is the first attempt to apply the double channel convolution neural network to HINs. We use it to train the structure information and content information of nodes simultaneously for similarity search, which avoids the problem that the content information embedded in the previous training of nodes decays with the following structure training.
- Two different attention mechanisms are designed in SSDCC+ to fully realize the personalization of the constructed model. First, an attention mechanism is used for multiple metapaths when embedding structure information, which flexibly and personally implements comprehensive consideration of structural information under different metapaths for each target node. Second, another attention mechanism is used between the content representation and the structure representation to combine them.
- We design a content and structure importance evaluation function based on the attribute information of the dataset to fuse the output results of two channels and propose a dual channel fusion module under the influence of attributes to improve the accuracy of search results and make the model more explainable.
- We have implemented our model on a public dataset to evaluate its performance and compared it with the existing methods. Through the analysis of the results, this model can deal with the problem of similarity search on HINs well and achieved higher performance than the existing methods.

The rest of this paper is organized as follows. Section 2 describes related works involved in this research. Section 3 introduces some definitions used in this paper and states our problem. Section 4 implements a comprehensive representation of multiple path instances between nodes. Section 5 presents a double channel-based search model to train the content and structure information of nodes. Section 6 designs an evaluation function and fuses two channels under the influence of attributes to improve the model.
Section 7 conducts experiments and analyzes the results. Section 8 summarizes the conclusions and the contributions of this paper.

2 Related work

In this section, we briefly review work in three areas that are closely related to ours: network embedding on HINs, double channel convolutional neural networks related research, and the top-k similarity search task on HINs.

2.1 Network embedding

In recent years, with the development of HINs [25], a large number of network embedding methods for HINs have emerged. Utilizing node embedding on HIN, the nodes’ representations are turned from high-dimensional sparse vectors to low-dimensional vectors while the relationship information between nodes is preserved. Some researchers use this kind of node vector to measure the similarity of nodes directly.

The deep learning-based method is currently a mainstream node embedding method [26–30]. In homogeneous information networks, DeepWalk [18] and TriDNR [20] implement node embedding based on random walks, NAIE [31] design an integrated autoencoder to learn a more informative representation of nodes by integrating structure and attribute information, and there are many studies using GNNs [32]. In HINs, the main research includes metapath2vec [17], HIN2vec [21], HAN [22], H2Rec [33], etc. Metapath2vec represents nodes based on random walks, and HIN2vec uses a neural network to learn the representation of nodes and metapaths. However, both of them only consider a single metapath; thus, they cannot capture rich path semantic information of nodes in HINs. HAN adopts a node-level attention mechanism and a semantic-level attention mechanism, and uses the latter to distinguish metapaths to obtain semantic information. However, its parameters are shared for all metapaths, resulting in the same metapaths attention scores for different nodes. H2Rec considers both homogeneous and heterogeneous information networks, but fuse their embedded information based on a simple sigmoid function.

The similarity search method based on the above models cannot flexibly adjust the search according to the network state, which means that a more appropriate model is needed.

2.2 Double channel convolutional neural networks

Double channel convolutional neural networks refer to a two-stream architecture that trains data in two streams simultaneously and fuses them. It has been widely studied and used in machine learning tasks related to image processing, and was first proposed in [34] for action recognition in videos. [35] applied double channel convolutional neural networks to an air quality measurement model, and they use each channel to train different parts of the environment images for feature extraction. MDCC-Net [36] designs multiscale double channel convolution feature fusion module. The double channel convolutional neural network is used to combine features of different scales in the same encoding layer, which fully extracts the detailed information of the original image and learns more tumor boundary information. In other studies, double channel convolution neural networks have also been used for polarimetric SAR image classification [37], image smoke detection [38], image reconstruction [39], power edge image recognition [40], hand-drawn sketch recognition [41], hyperspectral image classification [42], etc.

The various double channel convolutional neural networks designed in the above researches are mainly used to extract image features from different aspects. In this work, we attempt to use two channels to simultaneously capture the content information and structure information of nodes in the network, so existing models cannot be directly applied. We need to design a new framework suitable for the similarity search task on weighted HINs.

2.3 Top-k similarity search

There have already been many works to measure the similarity of objects in HINs. PathSim [12] uses metapaths and defines the similarity between two objects of the same type by considering the accessibility and visibility between vertices to solve the similarity search problem. HeteSim [23] is an extension of PathSim, which can measure the similarity of different types of objects. W-PathSim [13] proposes another improvement of PathSim by using the weighted cosine similarity of topics. RoleSim [43] considers both structural features and Jaccard similarity of attribute features. TopCPathSim [44] is proposed to address the problem related to topic-driven similarity search based on a constrained metapath. NeuPath [45] transforms PathSim computation to a deep learning model, which relieves the limitation of calculating large graph in traditional PathSim. SimRank [4] is a classical similarity search method on a homogeneous information network, and it assumes that two objects are similar if they are
referred by a similar object. Carmo [46] addresses the problem of set query with SimRank and applies it to personalized link prediction and event recommendation. CrashSim [47] enables computation of SimRank in both static networks and temporal networks. Similarity search is also applied to recommendation tasks and Holistic influence maximization tasks [48]. For example, MCRec [11] use an attention mechanism to measure the similarities between users and items. However, different users and items will jointly affect the weight of metapaths, which makes the model inefficient. In fact, only calculating the weight according to the user information to provide recommendations can also meet the needs.

The above approaches only consider a single aspect, such as content or structure similarity in HINs, and some only consider one metapath when measuring the structure similarity.

Some methods [6, 16] considering content and structure information usually embed the content information first and then embed the structure with the content embedding result as the initial input. However, similar to an optimization model seeking the minimum value, the initial state usually has little effect on the final result. In other words, as the model is trained by the nodes’ structure information, the content features contained in the initial representation will fade or even disappear. In addition, the above methods do not consider the time information in the network and treat the interaction at different times equally.

3 Preliminaries

In this section, we briefly introduce the preliminaries that will be used later and formulate the problem.

Definition 1 (Heterogeneous Information Network (HIN))

Given an information network, it can be represented by a directed graph \(G = (V, E, A, \mathcal{R}, \mathcal{A}; \phi, \varphi)\), where \(V\) is the object set, \(E\) is the link set, \(A\) is the set of link types, and \(\mathcal{R}\) is the set of link types, and \(\mathcal{A}\) is the attribute set corresponding to different types of links in \(\mathcal{R}\). In addition, \(\phi: V \rightarrow A\) is a mapping function from the object set to its type set, where each object \((v \in V)\) belongs to a specific object type \((\phi(v) \in A)\). \(\varphi: E \rightarrow \mathcal{R}\) is a mapping function from the link set to the link-type set, where each link \((e \in E)\) belongs to a specific link type \((\varphi(e) \in \mathcal{R})\). When the number of object types \(|A| > 1\) (or the number of link types \(|\mathcal{R}| > 1\)), this network is a heterogeneous information network (HIN).

In an HIN, if the attribute values are not all equal to 0, that is, \(\exists x, a^k_x \neq 0\) (\(a^k_x\) is the value of the attribute corresponding to edge \(x, al_x^k \in \mathcal{A}\)), the network is a weighted heterogeneous information network (weighted HIN).

Example 1 Considering an HIN on bibliographic data in Fig. 1, there are three types of nodes: authors, papers and conferences. Four types of links exist: writing and written-by between authors and papers, published-by and published-in between papers and the venues. The attribute position corresponding to writing and written-by represents the position of the author in the author list of the paper, and the attribute time corresponding to publishing and published-in stands for the year in which the paper was published in this venue.

We also use a metapath to express the rich path semantic information in an HIN.

Definition 2 (Metapath) A metapath expressed as \(A_0 \xrightarrow{R_0} A_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{l-1}} A_l\) is a path in an HIN, where \(A_0, A_1, \ldots, A_l \in A\) and \(R_0, R_1, \ldots, R_{l-1} \in \mathcal{R}\). It defines a combination relationship \(\mathcal{R} = \mathcal{R}_0 \circ \mathcal{R}_1 \circ \cdots \circ \mathcal{R}_{l-1}\) between object types \(A_0\) and \(A_l\). For simplicity, we directly represent a metapath as \(P = (A_0, A_1, \ldots, A_l)\). We call a specific path \(p = (a_0, a_1, \ldots, a_l|w_p)\) of a given metapath \(P\) a path instance, where \(w_p\) is the weight of path instance \(p\), and it can be calculated according to the attribute \(al \in \mathcal{A}\) of links \(\{(a_0, a_1), (a_1, a_2), \ldots, (a_{l-1}, a_l)\}\) on the path instance \(p\).

Definition 3 (Top-k similarity search in HIN) Given an HIN, in which \(G = (V, E, A, \mathcal{R}, \mathcal{A}; \phi, \varphi)\), its metapath set \(\{P_1, P_2, \ldots, P_p\}\), a query node \(n_q (n_q \in V)\), and the number of results \(k\). Top-k similarity search aims to find the most similar nodes of \(n_q\) from \(G\) based on a similarity measure method \(F\). How to define a proper \(F\) will be studied in this paper.

Fig. 1 Example of HIN on bibliographic data.
4 Capturing semantic information between nodes

The purpose of our research is to find the k nodes that are most similar to the nodes queried in the weighted HINs, so we need to calculate how similar it is to all other nodes in the network. In this section, we will implement a comprehensive representation of multiple path instances between two nodes, thereby capturing semantic information between nodes. First, we use the pretraining method to generate content and structure representations of nodes and extract their attribute information. Then, we generate inputs by sampling the path instances starting from the target node, which is the object to be queried, and design the neural network model to obtain the representation of the path instances. Finally, we propose a comprehensive path representation model with an attention mechanism between metapaths. Furthermore, to capture the rich information of HINs, we add an on-path attributes capture module during the path instance sampling process.

4.1 Node representation and attribute extraction

Node information in HINs can be divided into two parts, including content information and structure information, as shown in Fig. 2. The content information refers to text descriptions and other attribute information, such as images and labels, while the structure information refers to the connection relationship between nodes on the network.

First, we extract the content information of nodes. Specifically, it includes capturing nodes’ attribute information and representing text descriptions. For example, Fig. 2a shows the content information of a paper-type node. The abstract can be used for content embedding, and others can be used to extract attribute information. We capture the attribute information \( An = \{ an_1, an_2, \ldots, an_l \} \) of each node and use it in Sect. 4.2.3. Considering a node \( a \), we use Doc2vec [24] to generate its text representation \( a^c \in \mathbb{R}^{|A| \times d_1} \), where \( |A| \) is the number of nodes associated with type \( A \) and \( d_1 \) is the nodes’ content embedding dimension.

Second, we use another embedding method to represent nodes’ structure information. Many models have been proposed to embed nodes in HINs through metapaths because of the rich semantics they involve. In this research, we use metapath2vec++ [17] to generate nodes’ structure representation \( a^s \in \mathbb{R}^{r \times |A| \times d_2} \), where \( r \) is the number of metapaths, and \( d_2 \) is the nodes’ structure embedding dimension. Moreover, metapath2vec++ trains nodes based on a single given metapath. Therefore, for each metapath \( P_i \), we can obtain the nodes’ structure representation \( a^s_i \), where \( i \in \{1, \ldots, n\} \) denotes the given metapath id.

4.2 Path representation

The goal of path representation is capturing the rich semantic information between two nodes personally and flexibly and obtaining the path structure representation and the path content representation. Its framework is shown in Fig. 3.

4.2.1 Integrated embedding of nodes

In Sect. 4.1, the model generates a series of structure embeddings \( a^s_i \) corresponding to \( r \) given metapaths, and it also generates one content embedding \( a^c \) for each node. To map the vectors of nodes to a new space, we design three dense layers Dense 0, Dense 1 and Dense 2. We obtain the integrated structure embedding \( a^s \) of nodes by inputting \( a^s_i \) into Dense 1 and obtain the integrated embedding \( a^{s+c} \) of nodes including both structure and content by inputting \( a^s_i \) and \( a^c \) into Dense 0. We also input \( a^c \) into Dense 2 to maintain consistency.

![Fig. 2 Node representation and attribute information extraction](image)
4.2.2 Path instance and metapath representation

Many methods calculate nodes’ similarity in HINs by their embedding vector’s similarity directly. Inspired by the recommendation model MCRec [11], which uses path instances as input to capture the nodes’ interaction, we represent metapaths to capture rich path semantic information between nodes in similarity search tasks. As shown in Fig. 3, for the target node $a_1$, we take it as the starting point to sample the path instance according to the designed metapath. Under the metapath $P_i(i \in \{1, \ldots, n\})$, we obtain multiple path instances $p_{ij}(j \in \{1, \ldots, m\}$ is the id of a path instance) from $a_1$ to another node $a_2$ in the weighted HIN and calculate the path weight according to the attributes on the edge between adjacent nodes. In SSDCC+, we use a weight-based random walk to obtain path instances and record their weights. Specifically, in each time unit, it will take a step to one of its neighbours according to the transition probability, which comprehensively considers multiple attribute factors between nodes, such as time information. In addition, to reduce the training time, we abandon some path instances with low weight.

In a weighted HIN, different relationships on a path instance contain different attributes, such as the time when the author attended the venue and the author’s station of a paper. The values of these attributes can enrich the information of nearby links and nodes and reflect the data distribution characteristics of the whole dataset. A dynamically designed model according to data characteristics can make the model more explainable and improve the accuracy of the search result. Therefore, for a given metapath $P_i(i \in \{1, \ldots, n\})$, we capture attributes $A_{l_{ij}} = \{a_{l_{1}}, a_{l_{2}}, \ldots, a_{l_{l}}\}$ contained in each path instance $p_{ij}(j \in \{1, \ldots, m\})$. In Sect. 6, we will design the evaluation function and use $A_{l_{ij}}$ to improve the model.

During the pretraining process, the embedding of nodes on different path instances is generated corresponding to specified metapaths. Then, for nodes’ structure information, we obtain the embedding of path instance $p_{ij}(i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\})$ through convolutional layers. After that, the model obtains the embedding of metapath $P_i$ through a pooling layer. For nodes’ content information, we obtain the embedding of path instance $p_{ij}$ and the embedding of metapath $P_i$ in the same way.

4.2.3 Path representation

To obtain the comprehensive representation of the path containing all metapaths $P_i$ while distinguishing each target node, we apply an attention mechanism to fuse different metapaths. Considering that the path attention weight is affected by the target node, we use an attention scoring function with Bahdanau’s additive style to calculate the importance of each metapath $P_i$ to node $a_m(m \in \{1, \ldots, |A|\})$. The importance score $score(a_m, P_i)$ is calculated as follows:

$$score(a_m, P_i) = v_{s}^{T} \cdot \tanh(W_{1} \cdot a_{m} + W_{2} \cdot P_{i}) \tag{1}$$

where $W_{1}$ and $W_{2}$ are trainable weight matrices, $v_{s}^{T}$ is a trainable vector, and $P_{i} \in \{P_{i,1}^{c}, \ldots, P_{i, n}^{c}\}$ is the embedding of metapath $P_i$. After obtaining the importance score $score(a_m, P_i)$, the attention weights of $P_i$ can be calculated through the softmax function:

$$z_{a_m-P_i} = \frac{\exp score(a_m, P_i)}{\sum_{l=1}^{n} \exp score(a_m, P_{l})} \tag{2}$$

which can be interpreted as the contribution of the metapath $P_i$ for node $a_m$. $z_{a_m-P_i}$ indicates that node $a_n$ has a path to $a_m$ under metapath $P_i$. Based on the attention weights $z_{a_m-P_i}$ in Function 2, path representation can be computed as the weighted average of the metapath representation $P_i$. The content representation $C_{a_m}$ and structure representation $S_{a_m}$ of the path are as follows:
\[ S_{a_{m,n}} = \sum_{i} \mathbf{g}_{a_{m,n} \rightarrow P_i} \cdot P_i^S \quad (3) \]
\[ C_{a_{m,n}} = \sum_{i} \mathbf{g}_{a_{m,n} \rightarrow P_i} \cdot P_i^C \oplus an_1 \oplus an_2 \oplus \cdots \oplus an_l \quad (4) \]

where \( an_1, an_2, \ldots, an_l \) are the attributes of nodes on all path instances in \( P_i \).

### 5 Double channel-based search model and algorithm

The model is designed for similarity search tasks on a weighted HIN. To train the content information and structure information of each node, we design a double channel convolutional neural network, and the output is the content representation and structure representation of paths’ semantics between two nodes. In addition, to capture sufficient text description information, we add the attributes captured in the pretraining part to the content channel. We use another attention mechanism on the above two channels’ outputs to generate a comprehensive representation of the semantics between two nodes. Finally, we design the similarity search algorithm.

In this section, we propose the model for top-k similarity search based on a double channel convolutional neural network on weighted HINs (abbreviated as SSDCC+) and give the specific top-k similarity search algorithm. The model is divided into three parts: node representation, path representation, and the combination of content and structure. We will elaborate the details in the following sections.

#### 5.1 Double channel CNN-based path representation

The key challenge in SSDCC+ is how to comprehensively consider the content information and the structure information of the object. The existing methods \([49, 50]\) train the nodes’ structure information after content information, which can be considered to be a representation just for structure information. To comprehensively represent the content information and the structure information, we construct a double channel convolutional neural network-based method. The content representation and structure representation are trained by different channels called \textit{Cont-channel} and \textit{Stru-channel}, respectively, based on the double channel mechanism.

In \textit{Cont-channel}, nodes on the path instance \( p_{i,j} \) are represented by \( a^c \) described in Sect. 4.1. After path representation in Sect. 4.2, we acquire the content embedding \( C_{a_{m,n}} \) for given metapaths, which consider both text description and attribute information. Analogously, in \textit{Stru-channel}, nodes on the path instance \( p_{i,j} \) are represented by \( a^s \), and nodes on different metapaths \( P_i \) have their corresponding embedding \( a^s_i \). Finally, we acquire the structure embedding \( S_{a_{m,n}} \) from this channel.

#### 5.2 Combination of content and structure based on attention

From previous work, we obtain content embedding \( C_a \), structure embedding \( S_a \), and integrated embedding \( a^{c+s} \). In this section, we combine them by another attention machine, as shown in Fig. 4.

The attention score over different representations is:

\[ \text{score}(a^{c+s}, C_{a_{m,n}}) = v_{f_i}^T \cdot \text{tanh}(W_3 \cdot a^{c+s} + W_4 \cdot C_{a_{m,n}}) \quad (5) \]
\[ \text{score}(a^{c+s}, S_{a_{m,n}}) = v_{b_i}^T \cdot \text{tanh}(W_3 \cdot a^{c+s} + W_4 \cdot S_{a_{m,n}}) \quad (6) \]

where \( W_3, W_4 \) are trainable weight matrices and \( v_{f_i}^T \) is a trainable vector. Based on these two scores, the attention weights of content representation and structure representation are computed as follows:

\[ \alpha_C = \frac{\exp(\text{score}(a^{c+s}, C_{a_{m,n}}))}{\exp(\text{score}(a^{c+s}, C_{a_{m,n}})) + \exp(\text{score}(a^{c+s}, S_{a_{m,n}}))} \quad (7) \]
\[ \alpha_S = \frac{\exp(\text{score}(a^{c+s}, S_{a_{m,n}}))}{\exp(\text{score}(a^{c+s}, C_{a_{m,n}})) + \exp(\text{score}(a^{c+s}, S_{a_{m,n}}))} \quad (8) \]

Then, we acquire the combination of content and structure with the function shown as follows:

\[ CS = \alpha_C \cdot C_{a_{m,n}} \oplus \alpha_S \cdot S_{a_{m,n}} \quad (9) \]

In this function, we concatenate the results of two channels after multiplying by attention weights. Compared with the methods that summarize them together directly, such as other attention mechanisms, utilizing this function can effectively help us reduce information loss.

![Fig. 4 The attention mechanism for combining two channels](image-url)
5.3 Objective function and model training

We have obtained the comprehensive representation of content and structure information between target node \( a_m \) and its similar node \( a_n \). Furthermore, to carry out the top-k similarity search task, we need a similarity value between the two nodes. Therefore, we feed the representation \( CS \) into the MLP component to calculate the final output, that is:

\[
y_{m,n} = \text{sigmoid}[f(CS)]
\]

where \( f \) is the MLP component that has two dense layers with the ReLU activation function. Its output will be fed into a sigmoid layer to obtain the result \( y_{m,n} \).

We use a logarithmic loss function in the model. Therefore, \( y_{m,n} \) of node pair \((m, n)\) in which the node is similar to the other equals 1 while the dissimilar ones equal 0. The objective function is shown as follows:

\[
o = -\frac{1}{|N^+|} \sum_{i \in N^+} y_i \log p_i - \frac{1}{|N^-|} \sum_{i \in N^-} (1 - y_i) \log (1 - p_i)
\]

(12)

where \( N^+ \) is the positive sample set and \( N^- \) is the negative sample set.

5.4 Top-k search algorithm

In this section, we introduce the top-k search algorithm to obtain the first k most similar nodes of the target node by Algorithm 1. Specifically, we first input the HIN \( H \) and the metapath set \( \{P_1, P_2, ..., P_x\} \) to the SSDCC+ model. The model is trained until the loss function stabilizes or reaches the threshold of training times we set in advance. For the target node \( n_q \) \((n_q \in V)\), we can obtain a similar score to every node in \( H \). Then, other nodes are sorted according to the score \( y_{n_q,n_i} \) \((i = 1, 2, ..., |V|, n_i \in V\), representing each node in graph \( H \)). Finally, we take the nodes ranked in the first k position as the output of the search algorithm.

```
Algorithm 1 Top-k search
Require: Heterogeneous information network \( H \), metapath set \( P \), a query node \( n_q \), \( k \)
Ensure: Result \( R \)
1: Initialize model SSDCC+ by \( H \) and \( P \)
2: for \( i \) from 1 to \(|V|\) do
3:    Training SSDCC+ and generate \( y_{n_q,n_i} \) \((n_i \in V)\)
4: end for
5: for \( i \) from 1 to \(|V|\) do
6:    Sort \( n_i \) by \( y_{n_q,n_i} \)
7: end for
8: for \( j \) from 1 to \( k \) do
9:    Add \( j \)-th node into \( R \)
10: end for
```

where \( y_i \) is the output of the model, \( N \) is the input sample set, and \( p_i \) is the probability that the predicted input instance is similar.

Intuitively, a target node must have more path instances between it and its similar node under a given metapath, and the output \( y_{m,n} \) must be larger than others. Therefore, we use negative sampling to train the model where the negative dataset includes node pairs without path instance to \( a_m \) under the given metapath. We set the score of these dissimilar node pairs to zero, and then the objective function can be formulated as follows:

6 Dual channel combination under the influence of attributes

The method proposed in Sect. 5.2 uses an attention mechanism to fuse the dual channel’s output, which is dynamic and personalized, but the explainability is insufficient. According to the distribution characteristics of data in the network, it is more reasonable to analyse and use attribute information of data for dual channel fusion. In this section, we will design a function to evaluate the importance of \( \text{Cont-channel} \) and \( \text{Stru-channel} \) and propose a dual channel fusion model under the influence of attributes to
improve the accuracy of search results and make the model more explainable.

The attribute information is extracted during the path sampling process, as described in Sect. 4.2.2. To evaluate the importance of the two channels, we first need to determine which attributes affect it and how these attributes affect it.

6.1 The weight of time difference

First, in weighted HINs, we consider that the time difference \( \Delta t \in Al \) (\( Al \subseteq Al \) is a collection of attributes that contain attributes on the metapath \( P_i \) ) in metapath \( P_i \) is a key factor. Intuitively, for the path instance \( pi,j \) with two time attributes, the greater the difference between the two time attributes is, the greater the uncertainty of the information contained in it. Therefore, the importance of structural information between two nodes decreases, while the importance of content information increases. Under the influence of the time difference, the pivotal weights of \( pi,j \) 's structure information are as follows:

\[
\beta_{pi,j}^s = \frac{\beta_{min}^s - \beta_{max}^s}{t_{max} - t_{min}} \times \Delta t + \beta_{max}^s
\]  

where \( \Delta t \) is the time difference and \( t_{min} \) is the earliest time in the dataset, i.e., the minimum value of the time attribute, \( t_{max} \) is the latest time, \( \beta_{min}^s \) is the minimum value of \( \beta^s \) and \( \beta_{max}^s \) is the maximum value of \( \beta^s \).

Then, the comprehensive structure pivotal weight \( \beta^s \) between two nodes is calculated based on the abundant path instance information contained in multiple metapaths:

\[
\beta^s = f(\beta_{1,1}^s, \ldots, \beta_{i,j}^s, \ldots, \beta_{n,m}^s)
\]  

where \( n \) is the number of metapaths between two nodes, \( m \) is the number of path instances under the last metapath \( \{P_s\} \), and \( f \) is a statistical function suitable for the data distribution of the current weighted HIN. In this model, we simply design \( f \) as a function to calculate the average value of the input data. In addition, we should not ignore structural information because the time difference is too large or ignore content information because it is too small. Therefore, there is a restriction on the value range: \( 0 < \beta_{min}^s \leq \beta^s \leq \beta_{max}^s < 1 \).

6.2 The weight of path instances' number

We believe that another key factor affecting the importance of \( Cont-channel \) and \( Stru-channel \) is the number of path instances under a given metapath. Although it is not an attribute corresponding to a type of link, it reflects the characteristics of the data distribution in the network. The more path instances there are, the more active the node is in the network, and the more important the structure information is; in contrast, the content information is more important. Under its influence, the pivotal weights of structural information are as follows:

\[
\beta_{num} = \frac{\beta_{max}^{num}}{num_{max}} \times num
\]

where \( num \) is the number of path instances between two nodes, \( \beta^{num}_{max} \) is the maximum value of \( \beta^{num} \), and \( num_{max} \) is the maximum number of path instances we set according to the distribution of data. Theoretically, if the number of path instances is zero, it is not suitable to consider structure information because the path instance part of the model input is a zero vector, and it is meaningless to calculate the similarity with \( \theta \). Consequently, there is a restriction on the value range: \( 0 \leq \beta^{num}_{num} \leq \beta^{num}_{max} < 1 \).

6.3 Importance evaluation function

In this research, the influence of \( \beta^s \) and \( \beta^{num}_{num} \) on channel importance is considered equal. The structure and content pivotal weights under the influence of attributes are obtained as follows:

\[
\beta_C = \frac{\beta^s + \beta^{num}}{2}
\]

\[
\beta_S = 1 - \beta_C
\]

Combined with the results of Sect. 5.2, the output of the two channels is fused by considering both the attention mechanism and attribute importance. The specific function is changed from Function 9 to:

\[
CS_{atten+attr} = \alpha_C \cdot C_{att} + \beta_C \cdot C_{attr} + \alpha_S \cdot S_{att} + \beta_S \cdot S_{attr}
\]

In this function, we multiply the attribute weights \( \beta_C \), \( \beta_S \) with the structure representation and content representation and add them to the result of Function 9. This enhances the explainability and dynamics of the model.

According to the above analysis, we update the similarity score (Function 10) as:

\[
y_{m,n} = \text{sigmoid} [f(CS_{atten+attr})]
\]

In summary, the overall framework of SSDCC+ is shown in Fig. 5. The model is divided into five parts: Node Representation, Node Attribute Capture, Path Representation, Channel Importance Evaluation, and Combination of Content and Structure. Specifically, Sect. 4 carries out the node representation and attribute extraction, takes the result as the input of path representation, and obtains the single channel output containing semantic information. Section 5 proposes the dual channel combination function, which uses the attention mechanism to fuse the content and
structure information. Then, in Sect. 6, the importance evaluation function is proposed to improve the model, and the dynamic and reasonable nodes’ similarity score are obtained.

7 Experimental evaluation

In this section, we describe the datasets of our experiment, prove the accuracy of SSDCC+ and compare it with variants and baseline methods.

7.1 Environment and data collection

We use the Academic Social (AS) Network from AMiner [51, 52], which is publicly available, for the experiments. This network includes 2.1M papers, 8M citations, 1.7M authors and 4.3M coauthors. We construct a weighted heterogeneous graph based on three types of nodes: authors, papers, and venues. We also extract the attributes of different types of nodes, different relation edges among three sets of nodes (authors writing papers, papers published on venues, authors collaborating on a paper, and papers cited by other papers), and the attributes corresponding to specific edges to construct the network. We prune all papers that lack information on affiliations, year and publication venue.

The data of the original network AS are too large. We extracted two subsets, AS-5 and AS-small, for experiments. The AS-5 contains papers published from 2010 to 2014, their related authors, and conferences where papers

| Dataset statistics of AS-5 and AS-small |
|----------------------------------------|
| Dataset   | AS-5  | AS-small |
| Nodes     | 45226 | 7525     |
| Authors   | 20280 | 3334     |
| Papers    | 127   | 2        |
| Venues    | 125342| 13774    |
| Relations | 3431  | 1023     |
| Writing   | 60840 | 10002    |
| Publishing | 20280 | 3334     |
| Collaborating | 125342 | 13774 |    |
| Citing    | 3431  | 1023     |
were published. However, it is a sparse dataset containing a large number of unpopular authors and conferences. In order to alleviate the high computational cost and observe the experimental results on popular nodes, we further extract the data to construct another dataset AS-small, with similar extraction methods appearing in other studies [12, 53]. AS-small is a smaller dataset, screening conferences with at least 900 papers published between 2010 and 2014 and papers with at least one author contributing five papers. Compared with the first dataset, the nodes’ popularity on AS-small is higher, which means that it is easier to evaluate the similarity of nodes in manual annotation. The statistics can be found in Table 1. In particular, the number of nodes Paper in the table is the same as the number of relationships. This is mainly because each paper can only be published once in one venue.

\[
\text{Venue} \xrightarrow{\text{Publishing}} \text{Paper}
\]

7.2 Experimental setup — metapath and model parameters

7.2.1 Metapath selection

In academic social networks, we take authors as target nodes. Although similarity measurement methods such as PathSim [12] cannot synthesize multiple metapaths at the same time to measure node similarity, the metapath used in it is reasonable. It presents two metapaths, APA and APVPA, which represent the scenarios of coauthoring papers and publishing papers in the same venue, respectively. In addition, metapath APPA indicates that one author cited another author’s paper, which is also very important.

Since the AS-small dataset contains only 8 conferences and they have published a large number of papers, any two authors in AS-small have at least 1/8 of the probability of being connected through APVPA, which makes the training results poor. Therefore, metapaths AA, APPA, and APVPA are used to train and test the SSDCC model for similarity measurement of author-type objects on AS-5, and metapaths APA and APPA are used on AS-small.

7.2.2 Parameter settings

We select the text description of the nodes as the content information and training data by Doc2vec to obtain content embeddings. Then, metapath2vec is used to embed node structure information under different metapaths. We set the embedding dimension of both content and structure to 128, that is, \( d_1 = 128 \) and \( d_2 = 128 \). In addition, all parameters are initialized using a uniform distribution. Here, we set the model training epochs to 5 and the number of negative samples to 5. The minibatch Adam optimizer is selected to optimize these parameters, where the learning rate is set to 0.001 and the batch size is set to 512. Moreover, to make the method scalable, we use multiple processes to train the model. Specifically, 6 processes are used to train in parallel on the experimental machine, and the training speed is increased by 6 times.

7.2.3 Capture of attributes and weights

In Sect. 4.1, the model captures attributes of nodes and adds attributes to the Cont-channel to enrich the content information. For authors, the attribute includes the count of published papers, the total number of citations, the H-index of this author, etc.; for papers, the attribute includes affiliations, year, etc.

In Sect. 4.2.2, the model uses a weight-based random walk (RW) to obtain the path instances and calculate their weights. The weight between two adjacent nodes on the path instance is calculated based on the value of the corresponding attributes of the edges between them. For example, the value of the attributes between two papers is the time difference of their publication (the larger the time difference is, the smaller the weight is), the value of the attribute between an author and a paper is the author’s position, and so on. In the random walk process, the probability of walking from the current node to any of its neighbours is equal, but the weight between the passed node pairs will be recorded. Then, the weight of a path instance is the product of all the weights on it. After that, we only keep a few path instances with the greatest weights. The statistics of path sampling results on datasets AS-5 and AS-small are shown in Table 2.

| Metapath | Description                        | AS-5   | AS-small |
|----------|------------------------------------|--------|----------|
| APA      | Coauthoring papers                 | 320112 | 27548    |
| APPA     | Author quotes another’s paper       | 43669  | 8858     |
| APVPA    | Paper published in the same venue   | 3636308| 1607298  |
combine the two channels. SSDCC is our previously published model.

- **SSDCC-**: The attention mechanism between the content and structure channels is not considered, and they are directly connected.
- **SS+S**: A single channel model is used that only considers Cont-channel.
- **SS+C**: A single channel model is used that only considers Stru-channel.
- **Metapath2vec.apvpa** [17]: We use the metapath2vec algorithm here and sample the path according to the metapath APVPA to obtain the embedding of the node. According to the Euclidean distance between the vectors of different authors, a similarity search is carried out.
- **Metapath2vec.apa** [17]: We use the metapath APA for path sampling to obtain the embedding of nodes. Similarity search is carried out by Euclidean distance.
- **Doc2vec** [24]: In this method, nodes are embedded according to the content information of the object and then the top-k similar objects are searched for each target node by using node vectors.

In practice, we make use of Keras for all compared algorithms, and ReLU is used by all neural nets in the hidden layers.

### 7.3 Experimental results and comparison

In AS-5, we analyse the data from 2010 to 2014, select 10 popular authors (e.g. Jack J. Dongarra, P S Yu, Elisa Bertino, and Jiawei Han) and label them with similar authors according to their research direction, paper organization and publication to test the accuracy of similarity search results. In the AS-small cohort, we randomly selected 10 authors who had published at least one paper and processed the data in the same way as in the AS-5. We label each result object with a relevance score as 0-not relevant and 1-relevant.

#### 7.3.1 Search result analysis

The top-10 similarity search results on AS-small are shown in Table 3, and the results on AS-5 are shown in Table 4. The tables show the search results of four test subjects randomly selected on two datasets. The search results on AS-small are taken as examples.

W Pedrycz is a professor and Canada Research Chair in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. His main research directions include computational intelligence, fuzzy modelling and granular computing, knowledge discovery and data mining. A series of similar authors, including Duoqian Miao, Young Chel Kwun, Jin Han Park, etc., are all well-known scholars in related research fields. From 2010 to 2014, they all participated in the same conference with W Pedrycz, copublished papers with W Pedrycz, or cited W Pedrycz’s (cited by W Pedrycz) paper. As seen from the search results in the table, our experimental results are very reasonable.

#### 7.3.2 Overall comparison

We evaluated the precision and normalized discounted cumulative gain (NDCG) of the top-10 similarity search results and compared them with the baselines and the variants of our model, as shown in Table 5. (In Table 5, M. apvpa stands for Metapath2vec.apvpa, M. apa stands for Metapath2vec.apa and M. appa stands for Metapath2vec.appa.) It can be observed that the performance of SSDCC+ is better than that of SSDCC, which means that considering nodes’ attributes in content representation and paths’ attributes in dual channel fusion can effectively improve the accuracy of search results. The performance of SSDCC is better than that of SSDCC- because the use of an attention mechanism between the Cont-channel and Stru-channel is also beneficial to improve the accuracy. The results of SSDCC+, SSDCC and SSDCC- are also better than those of SS+S and SS+C, showing that the model’s

| Rank | M. Rabbani | Peng Shi | Jing Li | K. S. Chin |
|------|------------|----------|---------|------------|
| 1    | M. Aramoon Bajestani | David Zhang | Yong Wang | Kwong-Chi Lo |
| 2    | G. Baharian Khoshkhou | Xinping Guan | Jian-Hua Ge | Jendy P. F. Leung |
| 3    | J. Razmi | Bing Chen | Defeng Ren | Yanli Li |
| 4    | M. M. Ghotboddini | Xiaoping Liu | Yonglei Jiang | Jian Fa Tang |
| 5    | H. Rahimian | Kefu Liu | Jinfeng Tian | Xinggang Luo |
| 6    | Neda Manavizadeh | Tao Fang | Huaxia Chen | Jian-Bo Yang |
| 7    | Davoud Moshtaghi | Jinqing Liang | Bo Ma | Yuhua Qian |
| 8    | Z. Mobini | Z. Wang | Jianhua Ge | J. Y. Liang |
| 9    | R. Tavakkoli-Moghaddam | Xiaohui Liu | Liuhua Wang | Feng Feng |
| 10   | F. Jolai | Shaocheng Tong | Chensi Zhang | V. Leoreanu-Fotea |
performance can be improved by using the double channel convolutional neural network, which comprehensively considers the content information and structure information. As SSDCC, SSDCC- considers multiple metapaths while Metapath2vec.apvpa, Metapath2vec.appa and Metapath2vec.apa are trained based on only one metapath, the performance of SSDCC and SSDCC- and its variants are all better than that of the baselines. In addition, Doc2vec has the worst performance. The first reason is that the academic social network dataset is rich in structural information and relatively insufficient in content information. The second reason is that the Doc2vec model does not consider the attribute information of objects but only learns the representation of text descriptions.

7.3.3 Comparison of NDCG

To evaluate the performance of models under different $k$ values, we plot the NDCG value with respect to the $k$ values in Fig. 6a and b. The SSDCC+ model always achieves the best performance in all similarity search tasks. In most cases, the NDCG of SSDCC and SSDCC-, which consider both content information and structure information, is higher than that of other methods that only consider single-node information. Specifically, in AS-5, when $k$ is equal to 7, the NDCG value of SSDCC+ is 0.5905, which is greater than 0.4385 of SSDCC, 0.4167 of SSDCC-, 0.3129 of SS+S, 0.25 of SS+C, 0.1667 of Metapath2vec.apa, 0.0594 of Metapath2vec.apvpa, and 0.0556 of Doc2vec. It shows that the output of SSDCC+ is indeed optimal.

In addition, on AS-small and AS-5, the performance of Doc2vec is very poor, mainly because the nodes’ structural information in academic social networks is dense, while content information is scarce. Because only a single metapath is considered, the search performance of metapath2vec.appa and metapath2vec.apa on AS-small and that of metapath2vec.apvpa and metapath2vec.apa on AS-5 are relatively low.

7.3.4 Comparison of precision

The precision of SSDCC+ and the comparison model under different $k$ values is shown in Fig. 6c and d. Compared with NDCG, the precision of similarity search results is relatively low. In any case, SSDCC+’s performance on the two datasets is obviously the best. Overall, the precision of the search results on AS-small is better than that on AS-5. This is mainly because the AS-small is smaller, which makes the candidates for manual labelling clearer, and the model training results are more accurate.

Moreover, in both AS-small and AS-5, when the number of $k$ is less than 6, the precision of each model gradually becomes 0. The reason is that many methods cannot search for similar nodes.

| Rank | Don Towsley | W Pedrycz | Sushil Jajodia | Francky Catthoor |
|------|-------------|-----------|----------------|-----------------|
| 1    | Fletcher Wicker | Duoqian Miao | Micha Sharir   | Vasileios I. Kelefoursas |
| 2    | Xiaoqian Lu  | Young Chel Kwun | E. Welzl   | A. Kritikakou  |
| 3    | Jianning Mai | Jin Han Park | Adam Sheffer   | Vincent Loechner |
| 4    | Hui E. Zang  | Hyun Ju Cho | Csaba D. Töth | Franz Franchetti |
| 5    | Sae-Woong Bakr | Hongyun Zhang | Vladlen Koltun | James C. Hoe    |
| 6    | Ashwin Sridharan | Fatih Emre Boran | Shay Mozes | Peter A. Milder |
| 7    | Antonio Capone | Ying Yu  | Jie Gao | Rachid Seghir |
| 8    | Cem Ersoy  | Diyar Akay | Noga Alon | Benoit Meister |
| 9    | Ignas G. Niemeegers | F. Herrera | M. J. van Kreveld | Praveen Raghavan |
| 10   | Patrick P. C. Lee | Serkan Genç | Sivan Toledo | Hongwei Zhu |

| Table 4 Top-10 similarity search results of four popular authors on AS-5 |
|----------------------|---------------------|---------------------|
| Rank | Don Towsley | W Pedrycz | Sushil Jajodia | Francky Catthoor |
| 1    | Fletcher Wicker | Duoqian Miao | Micha Sharir   | Vasileios I. Kelefoursas |
| 2    | Xiaoqian Lu  | Young Chel Kwun | E. Welzl   | A. Kritikakou  |
| 3    | Jianning Mai | Jin Han Park | Adam Sheffer   | Vincent Loechner |
| 4    | Hui E. Zang  | Hyun Ju Cho | Csaba D. Töth | Franz Franchetti |
| 5    | Sae-Woong Bakr | Hongyun Zhang | Vladlen Koltun | James C. Hoe    |
| 6    | Ashwin Sridharan | Fatih Emre Boran | Shay Mozes | Peter A. Milder |
| 7    | Antonio Capone | Ying Yu  | Jie Gao | Rachid Seghir |
| 8    | Cem Ersoy  | Diyar Akay | Noga Alon | Benoit Meister |
| 9    | Ignas G. Niemeegers | F. Herrera | M. J. van Kreveld | Praveen Raghavan |
| 10   | Patrick P. C. Lee | Serkan Genç | Sivan Toledo | Hongwei Zhu |

| Table 5 Comparison on precision and NDCG of top-10 similarity search results |
|----------------------|---------------------|---------------------|
| SSDCC+ | SSDCC | SSDCC- | SS+S | SS+C | M.apvpa | M.appa | M.aa | Doc2vec |
|---|---|---|---|---|---|---|---|---|
| AS-small | ndcg.@10 | 0.7069 | 0.6724 | 0.5154 | 0.4892 | 0.4160 | – | 0.176 | 0.3136 | 0.1166 |
|      | prec.@10 | 0.38 | 0.36 | 0.23 | 0.18 | 0.25 | – | 0.02 | 0.1 | 0.03 |
| AS-5 | ndcg.@10 | 0.7423 | 0.6870 | 0.6269 | 0.4713 | 0.3055 | 0.25 | – | 0.2400 | 0.0593 |
|      | prec.@10 | 0.22 | 0.13 | 0.11 | 0.08 | 0.05 | 0.03 | – | 0.06 | 0.01 |
7.4 Discussion

In this section, we will analyze and discuss the experimental results in more detail. First, according to the results in Table 5 and Fig. 6, it can be seen that models that only consider one metapath, such as Metapath2vec.apvpa, perform worse than models that consider multiple metapaths. Therefore, it is necessary to capture the semantic information of multiple metapaths at the same time. Second, many objects have text description information. The text description information of some objects can reflect most of their characteristics, such as paper, while that of many other objects contains insufficient characteristics, such as author. The performance of Doc2vec in Fig. 6 is very poor on both datasets, confirming the above phenomenon. Therefore, similarity search using text information is not applicable to some types of objects. Third, when the dataset is large, the accuracy of the model is poor. Obviously, applying the model to a small dataset will obtain better results. Finally, observing Fig. 6, when \( k \geq 6 \), the performance on both datasets becomes acceptable. Therefore, it is better to provide users with more search results rather than fewer, such as fewer than 6 results.

8 Conclusion

In this paper, we provided a focused study on finding similar nodes in HINs, while related works include similarity search in homogeneous information networks and other downstream tasks, including search and recommendation in HINs. We analyze the problems of similarity search in HINs. To overcome this challenge, including adjusting different metapaths for different queries, flexibly integrating structure information and content information to comprehensively represent nodes, and considering the time factor and other data distribution information, we propose a model for top-\( k \) similarity search based on a double channel convolutional neural network on weighted HINs (SSDCC\(^+\)), which uses a CNN network with two channels to train the content information and structure information for the first time. We generate different structure and content embeddings for each node according to different metapaths, which are not available in previous studies, and extract the internal attribute information of the node. SSDCC\(^+\) flexibly considers multiple metapaths between two nodes and uses an attention mechanism to obtain the structure and content representation of semantics between nodes. Then, we propose another attention mechanism, which considers the output results of the two
channels and the attributes of the dataset to combine the node’s content and structure information for its comprehensive representation. The above two attention mechanisms fully realize the personalization of similarity search. Finally, the importance evaluation function we designed effectively improves the accuracy of similarity search results, makes the model more explainable, and improves the performance of SSDCC+

The experimental results on a public dataset show that SSDCC+ is superior to other comparison models in the accuracy of similarity search, mainly because it considers multiple metapaths, content and structure information and has the advantages of personalization and explainability. In conclusion, it proves the effectiveness of our proposed model, and our method achieved higher performance than existing methods. In future work, we plan to automatically identify different metapaths, consider the dynamic characteristics of the network, and add more attribute relationships.

Acknowledgements An earlier version of this paper was presented at the International Conference on DASFAA 2020. This work was supported by the National Natural Science Foundation of China (61902055, 62137001, U1811261), and Fundamental Research Funds for the Central Universities (N2117001).

Declarations

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work. There is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled Semantic Enhanced Similarity Search on Weighted HIN.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Abbasi A (2015) Reviewing academic social network mining applications In: International Conference on Information and Communication Technology Convergence, ICTC 2015, Jeju Island, South Korea, October 28-30, 2015. IEEE, pp 503–508, https://doi.org/10.1109/ICTC.2015.7354596
2. Keong BV, Anthony P (2011) Pagerank: a modified random surfer model In: 2011 7th International Conference on Information Technology in Asia, pp 1–6, https://doi.org/10.1109/CITA.2011.5998269
3. Cabrera-Vives G, Reyes I, Förster F, et al (2017) Deep-hits: rotation invariant convolutional neural network for transient detection CoRR abs/1701.00458 arXiv:1701.00458
4. Jeh G, Widom J (2002) Simrank: a measure of structural-context similarity In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada. ACM, pp 538–543, https://doi.org/10.1145/775047.775126
5. Shi C, Hu B, Zhao WX et al (2019) Heterogeneous information network embedding for recommendation. IEEE Trans Knowl Data Eng 31(2):357–370. https://doi.org/10.1109/TKDE.2018.2833443
6. Ma X, Wang R (2019) Personalized scientific paper recommendation based on heterogeneous graph representation. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2923293
7. Xie F, Chen L, Lin D et al (2019) Personalized service recommendation with mashup group preference in heterogeneous information network. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2894822
8. Shi C, Zhang Z, Ji Y et al (2019) Semrec: a personalized semantic recommendation method based on weighted heterogeneous information networks. World Wide Web 22(1):153–184
9. Xie F, Chen L, Ye Y, et al (2018) Factorization machine based service recommendation on heterogeneous information networks In: 2018 IEEE International Conference on Web Services, ICWS 2018, San Francisco, CA, USA, July 2-7, 2018. IEEE, pp 115–122, https://doi.org/10.1109/ICWS.2018.00022
10. Jiang Z, Liu H, Fu B, et al (2018) Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking In: Chang Y, Zhai C, Liu Y, et al (eds) Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018. ACM, pp 288–296, https://doi.org/10.1145/3159652.3159715
11. Hu B, Shi C, Zhao WX, et al (2018) Leveraging meta-path based context for top-N recommendation with a neural co-attention model. In: Guo Y, Farooq F (eds) Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018 ACM, pp 1531–1540, https://doi.org/10.1145/3219819.3219965
12. Sun Y, Han J, Yan X et al (2011) Pathsim: meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment 4(11):992–1003
13. Pham P, Do P, Ta CDC (2018) W-pathsim: novel approach of weighted similarity measure in content-based heterogeneous information networks by applying LDA topic modeling In: Nguyen NT, Hoang DH, Hong T, et al (eds) Intelligent Information and Database Systems - 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018. Proceedings. Part I, Lecture Notes in Computer Science, vol 10751 Springer, pp 539–549, https://doi.org/10.1007/978-3-319-75417-8_51
14. Jin J, Qin J, Fang Y, et al (2020) An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In: Gupta R, Liu Y, Tang J, et al (eds) KDD ’20: the 26th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, August 23-27, 2020. ACM, pp 75–84, https://doi.org/10.1145/3394486.3403050
15. Yin Y, Zheng W (2021) An efficient recommendation algorithm based on heterogeneous information network. Complex 2021. https://doi.org/10.1155/2021/6689323
16. Zhang C, Song D, Huang C, et al (2019) Heterogeneous graph neural network In: Teredesai A, Kumar V, Li Y, et al (eds) Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019,
17. Dong Y, Chawla NV, Swami A (2017) metapath2vec: scalable representation learning for heterogeneous networks In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017 ACM pp 135–144 https://doi.org/10.1145/3079983.308036

18. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations In: Mcaussay SA, Perlch C, Leskovec J, et al (eds) The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ‘14, New York, NY, USA - August 24 - 27, 2014 ACM, pp 701–710, https://doi.org/10.1145/2627330.2627372

19. Grover A, Leskovec J (2016) node2vec: scalable feature learning for networks In: Krishnapuram B, Shah M, Smolar AJ, et al (eds) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016 ACM, pp 855–864 https://doi.org/10.1145/2939672.2939754

20. Pan S, Wu J, Zhu X, et al (2016) Tri-party deep network representation In: Kambhampati S (ed) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016 IJCAI/AAAI Press, pp 1895–1901, http://www.ijcai.org/Abstract/16/271

21. Fu T, Lee W, Lei Z (2017) Hin2vec: explore meta-paths in heterogeneous information networks for representation learning In: Lim E, Winslett M, Sanderson M, et al (eds) Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017 ACM, pp 1797–1806, https://doi.org/10.1145/3132847.3132953

22. Wang X, Ji H, Shi C, et al (2019a) Heterogeneous graph attention network In: Liu L, White RW, Mantrach A, et al (eds) The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019 ACM, pp 2022–2032, https://doi.org/10.1145/3308558.3313562

23. Shi C, Kong X, Huang Y et al (2014) Hetesim: a general framework for relevance measure in heterogeneous networks. IEEE Trans Knowl Data Eng 26(10):2479–2492. https://doi.org/10.1109/TKDE.2013.2297920

24. Le QV, Mikolov T (2014) Distributed representations of sentences and documents In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014 JMLR Workshop and Conference Proceedings, vol 32 JMLR.org, pp 1188–1196, http://proceedings.mlr.press/v32/le14.html

25. Shi C, Li Y, Zhang J et al (2017) A survey of heterogeneous information network analysis. IEEE Trans Knowl Data Eng 29(1):17–37. https://doi.org/10.1109/TKDE.2016.2598561

26. Bonner S, Kureshi I, Brennan J et al (2019) Exploring the semantic content of unsupervised graph embeddings: an empirical study. Data Sci Eng 4(3):269–289. https://doi.org/10.1007/s41007-019-00097-5

27. Zhou J, Cui G, Hu S et al (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

28. Wu Z, Pan S, Chen F et al (2021) A comprehensive survey on graph neural networks. IEEE Trans Neural Networks Learn Syst 32(1):4–24. https://doi.org/10.1109/TNNLS.2020.2978386

29. Liu X, Tang J (2021) Network representation learning: a macro and micro view CoRR abs/2111.10772 arXiv:2111.10772

30. Peng Y, Choi B, Xu J (2021) Graph learning for combinatorial optimization: a survey of state-of-the-art. Data Sci Eng 6(2):119–141. https://doi.org/10.1007/s41019-021-00155-3

31. Chen J, Zhong M, Li J et al (2021) Effective deep attributed network representation learning with topology adapted smoothing. IEEE Transactions Cybern. https://doi.org/10.1109/TCYB.2021.3064092

32. Scarselli F, Gori M, Tsoi AC et al (2009) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80. https://doi.org/10.1109/TNN.2008.2005605

33. Shao Y, Liu C (2021) H2vec: homogeneous and heterogeneous network embedding fusion for social recommendation. Int J Comput Intell Syst 14:1303–1314. https://doi.org/10.2991/ijcis.210406.001

34. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos In: Ghahramani Z, Welling M, Cortes C, et al (eds) Advances in Neural Information Processing Systems, vol 27 Curran Associates, Inc, https://proceedings.neurips.cc/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf

35. Wang Z, Zheng W, Song C (2019b) Air quality measurement based on double-channel convolutional neural network ensemble learning CoRR abs/1902.06942 arXiv:1902.06942

36. Zheng S, Lin X, Zhang W et al (2021) Mdcc-net: multiscale double-channel convolution u-net framework for colorectal tumor segmentation. Comput Biol Med 130(104):183. https://doi.org/10.1016/j.compbiomed.2020.104183

37. Hua W, Wang S, Xie W, et al (2019) Dual-channel convolutional neural network for polarimetric SAR images classification In: 2019 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2019, Yokohama, Japan, July 28 - August 2, 2019, IEEE, pp 3201–3204, https://doi.org/10.1109/IGARSS.2019.8899103

38. Zhang F, Qin W, Liu Y et al (2020) A dual-channel convolutional neural network for image smoke detection. Multim Tools Appl. https://doi.org/10.1007/s11042-019-08551-8

39. Wang L, Liu X, Chen D et al (2020) ECT image reconstruction algorithm based on multiscale dual-channel convolutional neural network. Complex 2020. https://doi.org/10.1155/2020/4918058

40. Zhou F, Ma Y, Wang B et al (2021) Dual-channel convolutional neural network for power edge image recognition. J Cloud Comput 10(1):18. https://doi.org/10.1186/s13677-021-00235-9

41. Zhang L (2021) Hand-drawn sketch recognition with a double-channel convolutional neural network. EURASIP J Adv Signal Process 2021(7):73. https://doi.org/10.1186/s13634-021-00752-4

42. Yu H, Zhang H, Liu Y et al (2022) Dual-channel convolution network with image-based global learning framework for hyperspectral image classification. IEEE Geosci Remote Sens Lett 19(1):5–8, https://doi.org/10.1109/LGRS.2021.3130358

43. Jin R, Lee VE, Hong H (2011) Axiomatic ranking of network role similarity In: Apte´ C, Ghosh J, Smyth P (eds) Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011 ACM, pp 922–930, https://doi.org/10.1145/2020408.2020561

44. Pham P, Do P (2020) Topic-driven top-k similarity search by applying constrained meta-path based in content-based schema-enriched heterogeneous information network. Int J Bus Intell Data Min 17(3):349–376. https://doi.org/10.1504/IJBIDM.2020.109295

45. Xiao W, Zhao H, Zheng VW, et al (2021) Neural pathsim for inductive similarity search in heterogeneous information networks In: Demartini G, Zucon G, Culpepper JS, et al (eds) CIKM ’21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 – 5, 2021 ACM, pp 2201–2210, https://doi.org/10.1145/3450637.3482454

46. Wang Y, Feng Z, Chen L et al (2021) Efficient similarity search for sets over graphs. IEEE Trans Knowl Data Eng 33(2):444–458. https://doi.org/10.1109/TKDE.2019.2931901
47. Li M, Choudhury FM, Borovica-Gajic R, et al (2020) Crashsim: an efficient algorithm for computing simrank over static and temporal graphs In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp 1141–1152, https://doi.org/10.1109/ICDE48307.2020.00103

48. Cai T, Li J, Mian AS et al (2020) Target-aware holistic influence maximization in spatial social networks. IEEE Transactions Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.3003047

49. Severyn A, Moschitti A (2015) Learning to rank short text pairs with convolutional deep neural networks In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval Association for Computing Machinery, New York, NY, USA, SIGIR ’15, p 373–382, https://doi.org/10.1145/2766462.2767738

50. Sakamoto T, Gao X, Yavari E, et al (2017) Radar-based hand gesture recognition using i-q echo plot and convolutional neural network pp 393–395, https://doi.org/10.1109/CAMA.2017.8273461

51. Tang J, Zhang J, Yao L, et al (2008) Arnetminer: extraction and mining of academic social networks In: Li Y, Liu B, Sarawagi S (eds) Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008 ACM, pp 990–998, https://doi.org/10.1145/1401890.1402008

52. Tang J, Fong ACM, Wang B et al (2012) A unified probabilistic framework for name disambiguation in digital library. IEEE Trans Knowl Data Eng 24(6):975–987. https://doi.org/10.1109/TKDE.2011.13

53. Yu W, Lin X, Ge J, et al (2020b) Semi-supervised collaborative filtering by text-enhanced domain adaptation In: Gupta R, Liu Y, Tang J, et al (eds) KDD ’20: the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020 ACM, pp 2136–2144, https://doi.org/10.1145/3394486.3403264

54. Yu M, Zhang Y, Zhang T, et al (2020a) Semantic enhanced top-k similarity search on heterogeneous information networks In: Nah Y, Cui B, Lee S, et al (eds) Database Systems for Advanced Applications - 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24-27, 2020. Proceedings, Part III, Lecture Notes in Computer Science, vol 12114 Springer, pp 104–119, https://doi.org/10.1007/978-3-030-59419-0_7

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.