Supplementary Material for Community Search over Heterogeneous Information Networks via Weighting Strategy and Query Replacement

Fanyi YANG[1] , Huifang MA[1,2*] ,Weiwei GAO[1], Zhixin LI[2]
1 College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
2 Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin 541004, China

Abstract Recently, community search over Heterogeneous Information Networks (HINs) has attracted much attention in graph analysis, which aims to search for local communities containing query node. Although existing community search studies in HINs have proved effective in converting· heterogeneous graphs to homogeneous graphs via pre-defined meta-paths with consistent head and tail node types, two major limitations still exist. First, they fail to properly utilize the intermediate nodes to assign weights on the edges of the induced homogeneous graph, which is crucial for capturing rich semantics in the meta-path. Secondly, the query node plays an important role in mining local subgraphs related to user's interests in heterogeneous information networks. Existing methods perform well when a query node comes from the core region of the target community. However, they struggle with the query-bias issue and especially perform unsatisfactorily if the query node stands at the boundary region. To tackle these two limitations, we propose a novel Community Search via weighted strategy and Query Replacement over HINs model (CSQR), which models the intermediate node of the meta-path to get the better induced homogeneous graph, and replaces the original 'intractable' query node with new search-friendly query node. Specifically, we devise a new weighting assignment strategy for induced homogeneous graph, which can make reasonable use of intermediate nodes and assign weights in the induced homogeneous graph to reflect the semantic relationships between nodes. Then, we establish a new query node replacement strategy in the induced homogenous graph for local community detection. The original query node is replaced with the node that is closer to the query node and has a higher clustering tendency. Extensive experiments on three real datasets demonstrate the effectiveness of our proposed method.

Keywords Community search, Weighted strategy, Query replacement, Heterogeneous information networks

1 Introduction

Graphs (or networks) are prevalent in real-life applications for modeling structured data such as social graphs [1], document citation graphs [2], and neurobiol-ogical graphs [3]. Currently, network analysis mainly focuses on homogeneous information networks with same type of nodes and link relationships. For example, the collaboration network in Fig.2(a) is a classical homogeneous network in which nodes stand for authors and edges indicate collaborative relationships between authors. However, with the development of the network, the types of nodes and edges become more diversified. A single type of homogeneous network can no longer meet the needs of researchers to explore the information network, so Heterogeneous Information Network (HINs) arises at the right moment. Compared with homogeneous networks, HINs can effectively model and process complex and diverse data based on comprehensive structural information and rich semantic information. Therefore, it is significant for the analysis of heterogeneous information networks.

HINs are prevalent in various domains, including bibliographic information networks, social media, and knowledge graphs. Fig.1 illustrates an HIN of the DBLP network, which describes the relationship among entities of different types (i.e., author, paper, venue, and topic). In specific, it consists of five authors (i.e., a_1, ⋯, a_5), five papers (i.e., p_1, …, p_5), two venues (i.e., v_1 and v_2), and two topics (i.e., t_1 and t_2). The directed lines denote their semantic relationship. For example, the authors a_1 and a_2 have written the paper p_1, which mentions the topic t_1, published in the venue v_1.

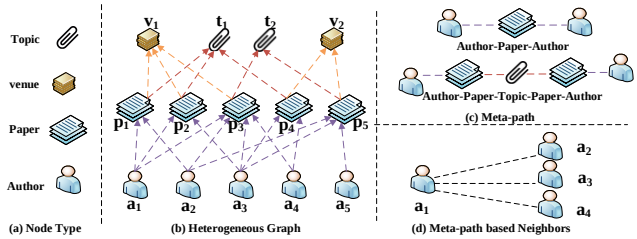


Fig.1: An illustrative example of a heterogeneous graph (DBLP). (a) Four types of nodes (i.e., author, paper, venue, topic). (b) A heterogeneous graph DBLP consists four types of nodes and three types of connections. (c) Two meta-paths involved in DBLP (i.e., Author-Paper-Author and Author-Paper-Topic-Paper-Author). (d) Author a_1 and its meta-path based neighbors (i.e., a_2, a_3 and a_4).

As a fundamental problem in network analysis, community search is widely used to find local community containing query node [4]. Different from community detection, community search pays more attention to local structure and user's personalized demand. Since the heterogeneous network has multiple types of objects and relationships and contains rich structure and semantic information, thus it has some nice features for community search on heterogeneous networks. For example, (1) it can find different types of communities, such as author communities, as well as venues, by using different meta-paths. (2) the query can be personalized, and different meta-paths indicate different relationships. By specifying different meta-paths for individual query vertices, we can obtain communities with different semantic relationships [5].

Despite the aforementioned advantages, community search in HINs remains a challenging task. Since HINs are networks with multiple typed objects and multiple typed links denoting different semantic relations, traditional community search methods cannot be directly applied to HINs. Moreover, the current methods for community search on HINs tend to convert HINs into homogeneous networks via defining meta-paths with consistent head and tail node type, and then a community search on homogeneous network is performed. But there are two limitations to this process. Firstly, given a meta-path, the HINs are mapped to the corresponding homogeneous information network by matrix multiplication. However, it is generally unreasonable to assign the edge weight of the homogeneous information network. For example, given the HINs in Fig.1(b) and the meta-path $\mathcal{P}$ =APA, taking a_2 as an example, the homogeneous network that can be obtained is shown in Fig.2(b), where the normalized cooperation frequency between two authors is used to describe the strength of the relationship between them. From the perspective of contribution to the paper, this kind of weight assignment is illogical. It is clear that a_1 contributes more to all papers of a_2 than a_3, so the strength of the relationship between a_1 and a_2 should be stronger than that between a_3 and a_2, which is not revealed in Fig.2(b). Secondly，the query node play essential roles in the community search effectiveness. Existing local community detection methods, such as the Random Walk with Restart (RWR)[6], perform well when the query node stand at the core region of the community. However, query-bias remains a vital issue [7]-[9]. Query-bias demonstrates that when a query node stands of the boundary region of the target community (i.e., the query node has connections with nodes outside its target communities), the detected community may miss some nodes that should belong to the target community or even include some nodes which may locate other community. Fig.2(c) gives the community search results for different query node with RWR. The red node is the query node and the nodes in the dashed boxes consist of communities. It can be seen that the nodes located at the core region of the community are able to obtain closely structured communities, while the boundary nodes influence the community search results because of their own properties.

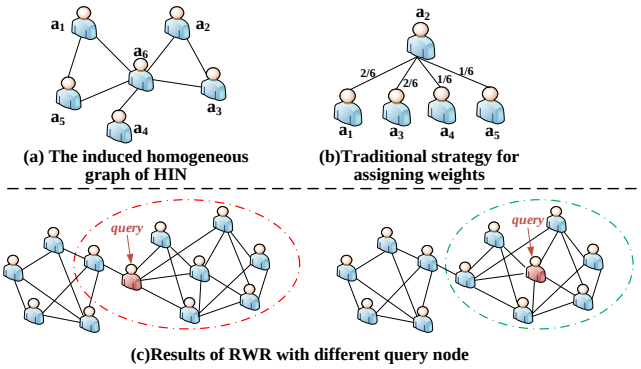


Fig.2: An illustrative example of an induced homogeneous graph and query-bias issue. (a) An induced homogeneous graph of Fig.1(b). (b) Traditional weight assignment strategy for v_2. (c) Local community detection results (with RWR) for different query node of a toy network.

To tackle the above challenges, in this work, we first propose a novel weighting assignment strategy on induced homogeneous graph, which makes reasonable use of intermediate nodes of meta-paths to model the relationship between two adjacent nodes in a meta-path instance, while combining the information of all meta-path instances under a given meta-path to capture richer semantic information. In addition, the information of a single meta-path may not provide a complete description of the whole network. In view of this, we fuse the information of multiple meta-paths so that the information between the meta-paths can complement each other. Secondly, to solve the query bias problem, we propose a new approach to detect the core nodes of the target community. The key idea is to find the node that is Close to the query node and have a high Local Clustering Coefficient around the query node (CLCC). For a query node, the node that is structurally close to the query node is obtained by using RWR, while its local clustering coefficients are examined in order to find the core nodes of the target community.

The contributions of this paper are summarized as follows:

- We propose a new weighting strategy to capture the rich semantic information in a meta-path by properly modeling the intermediate nodes of the meta path and integrating the information of multiple meta paths;
- We introduce a query node replacement strategy using CCLC score to avoid query bias in local community detection;
- We perform comprehensive experiments on three real-world networks to demonstrates the effectiveness of our proposed method.

The rest of our paper is organized as follows. Section 2 gives a brief review of related works on community search. Section 3 describes some preliminaries such as relatively problem definitions. Based on this, we give a detailed description about our method in Section 4. Then, we introduce the datasets, experimental settings and discuss the effectiveness of proposing algorithm in Section 5. Finally, we conclude this work and suggest future research directions in Section 6.

2 Related Works

Community can be loosely defined as the subsets of nodes which are more densely linked than the rest of the network. With the prevalence of networked systems, network analysis has been widely applied in practice. In this section, we introduce the most related two research topics community search on homogeneous graph and heterogeneous graph.

2.1 Community search in homogeneous graph

Community search has been studied extensively on homogeneous graph, which aims to find a connected subgraph for a network given a set of sample nodes. Existing methods can be roughly divided into two categories: based on cohesiveness metrics [10], [11], [12] and based on random walk theory [6], [13]. We briefly review two kinds of methods separately. Firstly, to measure the structure cohesiveness of a community，diverse community models have been proposed，For example, Sozio et al [10] develop an optimum greedy algorithm based on minimum degree and distance constraints. Moreover, Huang et al [11] study community search using the *k*-truss and formulate their problem as finding a Closest Truss Community (CTC). CTC aims to search a connected *k*-truss subgraph with the largest *k* that contains sample nodes. The pre-defined subgraph pattern imposes a very rigid requirement on the topological structure of community, which may not

perfectly hold in real-world community. Secondly, random walk-based algorithms are widely used because of their ability to find communities that are closely connected to the query node. For example, Tong et al. proposed the RWR[6], which improves the traditional random walk algorithm by obtaining an importance ranking vector against other nodes of the query node, and the vector can largely improve the quality of community search results. Andersen et al. proposed a PageRank-Nibble[13] algorithm and demonstrated the feasibility of conductance in community search. A comprehensive survey of community search models and existing approaches can be found in [4,9]. However, all these works focus on homogeneous graph, and it is not clear how to adapt them for community search over HINs.

2.2 Community search in heterogeneous graph

Heterogeneous information networks have many new characteristics, such as different types of objects and relationships. Meanwhile, HINs contain rich semantic information that can be captured by meta-paths. These new characteristics bring many new challenges to the task of community search on heterogeneous information networks. Nevertheless, some related works on community search in HINs have emerged in recent years. Jian[14] propose the relational community which is defined upon relational constraints. Using these constraints, the user can specify fine-grained requirements on vertex degrees. Fang[5] et al. use the well-known concept of meta-paths to model the relationship between two vertices of the same type, then measure the cohesiveness of the community by extending the classic minimum degree metric with a meta-path. The difference between the above two approaches is that the former finds communities that contain multiple types of nodes, while the latter aims to generate communities with nodes of the same type as the query node. Our work is more related to the latter as we find communities where vertices are of the same type. The main difference between this method and our method is that it performs community search based on the cohesion metric, which is inflexible and always too loose or too tight for the topology of the community.

3 Preliminaries

In this section, we give formal definitions of some important terminologies related to HINs. Graphical illustrations are provided in Fig.1. Besides, Table 1 summarizes frequently used notations in this paper for quick reference.

Definition 1 Heterogeneous Information Network. A heterogeneous information network is defined as graph $G=(V,E)$,where V and E represent the node set and the link set, respectively. In a HIN, each node v and edge e are associated with their type mapping functions $\phi(v):V\to\mathcal{A}$ and $\varphi(e):E\to\mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the sets of predefined node types and link types, respectively, where $|\mathcal{A}|+|\mathcal{R}|>2$.

Example. As shown in Fig.1(b), we construct a heterogeneous graph to model the DBLP. It consists of multiple types of node (author, paper, venue, and topic) and relations. The directed lines denote their semantic relationship. For example, the authors a_1 and a_2 have written a paper p_1, which mentions the topic t_1, published in the venue v_1.

In heterogeneous graph, two nodes can be connected via different semantic paths, which are called meta-path.

Definition 2 Meta-path. A meta-path $\mathcal{P}$ is defined as a path in the form of $A_1\xrightarrow{R_1}A_2\xrightarrow{R_2}\cdots\xrightarrow{R_l}A_{l+1}$ (abbreviated as $A_1A_2\cdots A_{l+1}$), which describes a composite relation $R=R_1\circ R_2\circ\cdots\circ R_l$ between objects A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations.

Example. As shown in Fig.1(c), two authors can be connected via multiple meta-paths, e.g., Author-Paper-Author(APA) and Author-Paper-Topic-Paper-Author (APTPA). Different meta-paths always reveal different semantics. For example, the APA indicates the co-author relationship, while Author-Paper-Topic-Paper-Author

Table 1 Main symbols and their definitions

Symbol	Definition
$\mathcal{P}$	Meta-path
$N^{\mathcal{P}}$	Meta-path based neighbors

$p(v_h, v_t)$	A metapath instance connecting node v_h and v_t
d_i	the degree of node v_i
$\mathbf{P}$	transition matrix
K	sliding window length
$\mathbf{m}^{(t)}$	score vector of v_q at time point t
$\mathbf{v}^{(t)}$	visiting history vector at time point t
$\mathbf{e}^{(t)}$	vector for key positions at time point t
C	community

(APTPA) represents the co-topic relation.

Definition 3 Meta-path Instance. Given a meta-path $\mathcal{P}$ of a heterogeneous graph, a meta-path instance p of $\mathcal{P}$ is defined as a node sequence in the graph following the schema defined by $\mathcal{P}$.

Example. Considering the meta-path APTPA in Fig.1, nodes a_1 and a_3 are connected via the meta-path instance a_1-p_1-t_1-p_2-a_3. Moreover, we may refer to p_1, t_1 and p_2 as the intermediate nodes along this meta-path instance.

Definition 4 Meta-path based Neighbors. Given a node i and a meta-path $\mathcal{P}$ in a heterogeneous graph, the meta-path based neighbors $N_i^{\mathcal{P}}$ of node i are defined as the set of nodes which connect with node i via meta-path $\mathcal{P}$. Note that the node's neighbors do not include itself.

Example. Taking Fig.1(d) as an example, given the meta-path APA, the meta-path based neighbors of a_1 includes a_2, a_3 and a_4. Similarly, the neighbors of a1 based on meta-path APTPA includes a_2, a_3, a_4 and a_5. Obviously, meta-path based neighbors can exploit different aspects of structure information in heterogeneous graph. We can get meta-path based neighbors by the multiplication of a sequences of adjacency matrices.

4 Methodology

In this section, we introduce the proposed CSQR model which is composed of three major components:(1) The induced homogeneous graph by meta-path; (2) Query node replacement; and (3) Local clustering by conductance.

4.1 The induced weighted homogeneous graph

In this paper, we focus on searching community in HINs, in which nodes are with a specific type (e.g., a community of authors in the DBLP network). Although HINs contain multiple types of nodes and relationships, there may be no connection between nodes of the same type. In order to connect two nodes of the same type, we adopt the well-known concept of meta-path, or a sequence of relations defining a composite relation between its starting type and ending type. At the same time, the edge weight between the head and the end nodes of the meta-path is deliberately calculated by considering the intermediate nodes of the meta-path.

Specifically，Given a HIN and a meta-path $\mathcal{P}=A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$, the probability between any two adjacent nodes under the meta-path $\mathcal{P}$ is formalized as follows:

$$Prob(v_i, v_j \mid \mathcal{P}) = \begin{cases} \dfrac{1}{|N_{t+1}(v_i)|} & (v_i, v_j) \in E, v_i \in A_t, v_j \in A_{t+1} \\ 0 & (v_i, v_j) \notin E \end{cases} \quad (1)$$

where $N_{t+1}(v_i)$ denotes the A_{t+1} type of neighborhood of node v_i. The pattern of the meta-path is repetitively followed until it reaches the pre-defined length.

Let $p(v_h, v_t)$ be a meta-path instance of the meta-path $\mathcal{P}$ from the head node v_h to the tail node $v_t \in N_{v_h}^{\mathcal{P}}$. $Prob(p(v_h, v_t))$ represents the probability of connecting edges between node v_h and node v_t based on this meta-path instance, we have

$$Prob(p(v_h, v_t)) = \prod_{v_i, v_j \in p(v_h, v_t)} Prob(v_i, v_j \mid \mathcal{P}) \quad (2)$$

$Prob(p(v_h, v_t))$ contains information about the intermediate nodes of the meta-path, which reflects the weights between nodes v_h and v_t in the induced homogeneous graph.

Since there are multiple meta-path instances of a given meta-path, the probability of a connection between node v_h and node v_t is the sum of the probabilities of all meta-path instances of meta-path $\mathcal{P}$

$$Prob^{\mathcal{P}}_{(v_h,v_t)} = \sum_{p(v_h,v_t)\in P} Prob(p(v_h,v_t)) \quad (3)$$

where P represents the set of all meta-path instances p of meta-path $\mathcal{P}$.

For example, given node a_2 and meta-path $\mathcal{P}$ =(APA), its induced weighted homogeneous graph is built as shown in Fig.3. Fig.3(a) gives all meta-path instances of node a_2 under the meta-path APA and weighted by the method in this paper. Fig.3(b) shows the connected edges of a_2 in the induced homogeneous graph. Compared with the traditional weighting strategy in Fig.2(b), the method in this paper can reasonably capture the semantic information among the nodes in HIN.

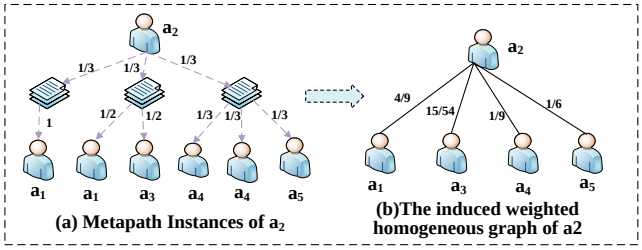


Fig.3 An illustrative example of a induced weighted homogeneous graph using our method. (a) An example of weighted element path of a_2 (b) An inductive homogeneous graph of a_2.

HINs have a wide variety of node types and links, and contain a wealth of semantic information. Different meta-paths in a HINs contain different semantic information. Therefore, the semantic information provided based on a single meta-path may not be able to take into account the diversity of information in the whole network. Intuitively, we believe that fusing multiple meta-paths can make better use of the network information.

Let there be N meta-paths $\mathcal{P}=\{\mathcal{P}_1,\mathcal{P}_2,...,\mathcal{P}_N\}$ in the network, and first obtain a weighted homogeneous graph $G_{\mathcal{P}_i}$ based on each meta-path according to the above method, and then integrate the weighted homogeneous graphs of different meta-paths to obtain the final induced weighted homogeneous graph $G_{\mathcal{P}}$. The final connection weights between two nodes in the induced weighted homogeneous graph can be obtained according to Eq(4)

$$Prob(v_h,v_t) = \frac{1}{N}\sum_{i=1}^{N} Prob^{\mathcal{P}_i}_{(v_h,v_t)} \quad (4)$$

Noted that algorithm 1 gives the process of generating a weighted homogeneous graph given a meta-path. Since there are multiple meta-paths with different semantics in the heterogeneous information network, in order to make full use of the semantic information, the above algorithm flow is used for each meta-path to obtain the corresponding homogeneous graph, and then all the homogeneous graphs based on different meta-paths are fused using Equation (4) to obtain the weighted homogeneous graph that is desired.

Algorithm 1: The induced weighted homogeneous graph

Input: A heterogeneous graph $G=(V,E)$,a metapath $\mathcal{P}_i$

Output: The induced weighted homogeneous graph $G_{\mathcal{P}_i}$

1: Collect the set S of nodes with the target type;
2: **for** each node $v \in S$ **do**:
3: Initialize a set X={v};
4: **for** $i \leftarrow 1$ *to length*($\mathcal{P}_i$) **do**:
5: $Y \leftarrow \emptyset$;
6: **for** each node $u \in X$ **do**:
7: **for** each neighbor t of u **do**:
8: **if** (u,t) matches with i-th edge of $\mathcal{P}$
9: **then** Y.add(t);
10: **calculate** *Prob*(u,t) based on Equation (1);
11: $X \leftarrow Y$;
12: **for** each node $u \in X$ ***do:***
13: add an edge between v and u;
14: **calculate** $Prob^{\mathcal{P}}_{(v,u)}$ based on Equation (3);
15: **return** $G_{\mathcal{P}_i}$

4.2 Query node replacement

In this section, a new query node replacement strategy is designed for the query bias problem to find the core nodes of the community. Noted that the subsequent contents are based on the weighted homogeneous graph obtained in section 4.1.

Intuitively, the core node of the community should have the following characteristics :(1) The core node is closer to the query node;(2) The core node should have a high clustering tendency. Based on above properties, we define the CLCC score of a node. The CLCC score depict the possibility of the node in the graph becoming the core node in the community. A node with the close

to the query node and closely connected around has a higher CLCC score and is more likely to become a core node.

Specifically, given a node, its CLCC score is shown in Equation (5):

$$CLCC(v_i) = \frac{d_i}{\max_{1\le j\le |V|} d_j} \times \frac{2\times \sum_{j,k\in N(v_i)} A_{jk}}{d_i(d_i-1)} \times \mathbf{m}_i \quad (5)$$

where $N(v_i)$ is the neighbors set of node v_i, and d_i is the degree of v_i. $\mathbf{m}_i$ is the mass of node v_i. Traditional measures of clustering tendency only take into account the closeness of a given node's neighbors and omit the effect of the node's own degree, which leads to erroneous amplification of the clustering tendency of a node with a small degree and closely connected neighbors. Therefore, in Equation 5, the degree of the node itself is taken into account to address this problem. It is observed that node with a large degree and closely connected neighbors and close to the query node is more likely to be the core node of the community.

Next, we discuss how to obtain $\mathbf{m}_i$. $\mathbf{m}_i$ depicts the closeness between core node and query node, which is obtained through Memory-based Random Walk(MRW) [15]. The key to the MRW approach is to use a sliding window to memorize the key positions that the walker has previously visited and aggregate the entire visiting history of the walker. In particular, the next step of the walker depends not only on the current visiting probability but also its previous visiting history. Specifically, in MRW，the visit probability of node t+1 is defined as：

$$\mathbf{m}^{(t+1)} = \alpha \mathbf{P}^{\top}\mathbf{m}^{(t)} + (1-\alpha)\mathbf{v}^{(t)} \quad (6)$$

where **P** is the column-normalized probability transfer matrix, which can be obtained according to Equation 4, $\mathbf{v}^{(t)}$ represents the aggregated history of the previous steps.

At time *t*, the key positions of the walker is defined as the node(s) with the largest visiting probability, and these key positions are represented by the vector $\mathbf{e}^{(t)}$. More specifically, suppose that there are $n \geqslant 1$ key positions, the entries in $\mathbf{e}^{(t)}$ corresponding to the n key positions are set to $1/n$, and all other entries are 0.

To record the visiting history, a sliding window of length *K* is used to aggregate these key positions. That is:

$$\mathbf{v}^{(t)} = \left(1-\beta^{t-1}\right)\mathbf{v}^{(t-1)} + \beta^{t-1}\frac{1}{K}\sum_{k=1}^{K}\mathbf{e}^{(t+1-k)} \quad (7)$$

where $\frac{1}{K}\sum_{k=1}^{K}\mathbf{e}^{(t+1-k)}$ represents the average of the key position vectors in the current time window, i.e., the past *K* steps. $\mathbf{v}^{(t)}$combines the previous visiting history (represented by $\mathbf{v}^{(t-1)}$) and the average of key position vectors in the current time window with a decay factor. Note that, initially, when $t<K$, $\mathbf{e}^{(t-K)}$ is set to be $\mathbf{e}^{(0)}$ and $\mathbf{v}^{(0)}= \mathbf{e}^{(0)}$, where $\mathbf{e}^{(0)}$ is the same as the vector **q** in RWR which represents the query node **q**.

Algorithm 2 gives the process of query node replacement. Noted that, we find the core node of the community in the third-order neighbors of the query node, as suggested on line 16.

Algorithm 2：Query node replacement strategy

Input：The induced weighted homogeneous graph $G_{\mathcal{P}}=(V_{\mathcal{P}},E_{\mathcal{P}},\mathbf{W}_{\mathcal{P}})$,transition matrix **P**,query node v_q, the time window *K*

Output：New query node v_{new}

1: $\mathbf{m}^{(0)}=\mathbf{0}$; $\mathbf{m}^{(0)}(v_q)=1$;
2: Init. uniform values for key posi. in $\mathbf{e}^{(0)}$ and $\mathbf{v}^{(0)}$
3: **while** no convergence **and** t<T **do**:
4: $\mathbf{m}^{(t+1)}=\alpha\mathbf{P}^{\top}\mathbf{m}^{(t)}+(1-\alpha)\mathbf{v}^{(t)}$;
5: compute $\mathbf{e}^{(t+1-k)}$ $(1\le k\le K)$;
6: update $\mathbf{v}^{(t+1)}$ based on equation (7);
7: **end while**
8: obtain the set $N(v_q)$ of neighbors of v_q by the adjacency matrix of $G_{\mathcal{P}}$, $v_{candidate} = v_q$, iterations = 0, $N_{all} = N(v_q)$;
9: **while** true:
10: $N_{temp}= \varnothing$， iterations += 1；
11: **for** all $v_i \in N_{all}$ **do**:
12: $N_{temp}=N_{temp}\cup N(v_i)$
13: **if** CLCC(v_i)>CLCC(v_q):
14: $v_{candidate}=v_i$
15: **end for**
16: **if** iterations>3*:*
17: break
18: **else***:*
19: $N_{all}=N_{temp}\cup (N(v_{candidate})-N(v_q))$
20: $v_q= v_{candidate}$
21: **end while**
22: **return** $v_{candidate}$

4.3 Local clustering by conductance

In order to discover local community with better quality, we use core node after node replacement to obtain score vector by performing a random walk with restart on the induced weighted homogeneous graph. For score vector, we first find nodes with the top-L largest scores. We set the default value of L to be 200 since most communities in real-world datasets are not very large [16]. Let $\{l_i\}(1 \le i \le \mathrm{L})$ represent the list of top-L nodes sorted in descending order. For each i $(1 \le i \le \mathrm{L})$, we compute the conductance[17] of the subgraph induced by node set $\{l_1,...,l_i\}$. The node set with the smallest conductance will be returned as the target community. Conductance measures the cohesiveness of a set of nodes C. The set of nodes with the smallest conductance is returned as the target community.

$$Conductance(C) = \frac{cut(C,\bar{C})}{\min\{vol(C), vol(\bar{C})\}} \tag{8}$$

where $\bar{C}$ is the residual set of C, $cut(C,\bar{C}) = \sum_{i\in C, j\in\bar{C}} A(i,j)$, $vol(C) = \sum_{i,j\in C} A(i,j)$.

Conductance measures the cohesiveness of a set of nodes, a small conductance(C) indicating that the set C is more closely connected internally and more sparsely connected externally.

5 Experiments

We implement our method in Python, and all the experiments are implemented on a computer with a 2.70 GHz CPU and 32 GB memory. In this section, we guide experiments on real-world datasets to evaluate our approach via answering the following research questions:

- RQ1: How does our proposed method perform as compared with state-of-the-art methods?
- RQ2: How is the quality of the communities found by our method?
- RQ3: CSQR mainly consists of the induced weighted homogeneous graph and query node replacement. How much does each component contribute?

5.1 Dataset description and evaluation metrics

5.1.1 Dataset description

We adopt three widely used heterogeneous graph datasets from different domains to evaluate the performance of CSQR as compared to state-of-the-art baselines. The detailed descriptions of the heterogeneous graph used here are shown in Table 2.

DBLP[1]: We extract a subset of DBLP which contains 14328 papers (P), 4057 authors (A), 20 conferences (C), 8789 terms (T). The authors are divided into four areas: Database, Data mining, Machine learning, Information retrieval. Also, we label each author's research area according to the conferences they submitted. Author features are the elements of a bag-of-words represented of keywords. Here we employ the meta-path set {APA, APCPA, APTPA} to perform experiments.

IMDB[2]: We extract a subset of IMDB which contains 4780 movies (M), 5841 actors (A) and 2269 directors (D). The movies are divided into three classes (Action, Comedy, Drama) according to their genre. Movie features correspond to elements of a bag-of-words represented of plots. We employ the meta-path set {MAM,MDM} to perform experiments.

Last.fm[3]: It is a music website keeping track of users' listening information from various sources. We adopt a dataset released by HetRec 2011[18], consisting of 1892 users, 17632 artists, and 1088 artist tags after data preprocessing. We employ the meta-path set {UAU, UATAU} to perform experiments.

5.1.2 Baselines and evaluation metrics

In order to measure the performance of the methods, the following two kinds of methods are selected for comparison. First, to compare the contributions of the weighting strategy and node replacement strategy proposed by the our methods, the following variants of the methods are used: CSQR-W indicates that the traditional weighting strategy is adopted; CSQR-R

[1] https://www.imdb.com/
[2] https://dblp.uni-trier.de/
[3] https://www.last.fm/

indicates that the node replacement strategy is omitted; and CSQR-WR indicates that neither is considered. Second, compared with existing community search methods on heterogeneous information networks, the Basic-core method proposed by Fang[5] et al. is selected, which is the first community search method on heterogeneous information networks so far. The comparison methods are described in detail as follows.

Basic-core[5]:The method first obtains the homogeneous graph from the HINs, and then obtains the community structure by the cohesiveness measure k-core.

CSQR-W: CSQR without using weighting strategy based on intermediate node of meta-path.

Table 2: Statistics of datasets

Dataset	Node	Relations	Meta-paths	Communities
IMDB	# Movie(M): 4,278 # Director(D): 2,081 # Actor(A): 5,257	# M-D: 4,278 # M-A: 12,828	MDM MAM	3
DBLP	# Author(A): 4,057 # Paper(P): 14,328 # Topic(T): 7,723 # Venue(V): 20	# A-P: 19,645 # P-T: 85,810 # P-V: 14,328	APA APTPA APVPA	4
Last.fm	# User(U): 1,892 # Artist(A): 17,632 # Tag(T): 1,088	# U-A: 92,834 # A-T: 23,253	UAU UATAU	5

CSQR-R: Query node replacement strategy of CSQR removed.

CSQR-WR: An extremely simplified version of CSQR, which combine the above two variants means removing both weighting strategy and query node replacement strategy.

Evaluation metrics. Recall, precision and F1-score are three common evaluation criterion for local community detection algorithms. The definitions of recall, precision and F1-score are as follows:

$$recall = |C_F \cap C_T| / |C_T| \tag{9}$$

$$precision = |C_F \cap C_T| / |C_F| \tag{10}$$

$$F1-score = \frac{2 \times precision \times recall}{precision + recall} \tag{11}$$

where C_T is the node set in real local community where the given node locates; C_F is the node set in the detected local community. From above, the recall is defined as the fraction between the number of common nodes in $C_T \cap C_F$ and the number of nodes in C_T. The precision is defined as the fraction of the number between common nodes in $C_T \cap C_F$ and the number of nodes in C_F . F1-score is the geometric mean of recall and precision. The value range of the above three evaluation indexes is between 0 and 1, and the larger the value is, the better the algorithm performance will be.

Normalized mutual information NMI, based on confusion matrix $\boldsymbol{M}$, is defined as follows:

$$NMI = \frac{-2\sum_{i=1}^{|C_T||C_F|}\sum_{j=1} M_{ij} \log\left(M_{ij}\boldsymbol{M} / \boldsymbol{M}_{i.}\boldsymbol{M}_{.j}\right)}{\sum_{i=1}^{|C_F|}\boldsymbol{M}_{i.} \log\left(\boldsymbol{M}_{i.} / \boldsymbol{M}\right) + \sum_{j=1}^{|C_F|}\boldsymbol{M}_{.j} \log\left(\boldsymbol{M}_{.j} / \boldsymbol{M}\right)} \tag{12}$$

where $M_{i,j}$ represents the number of nodes belonging to the real community C_i and the detected community C_j, $\boldsymbol{M}_{i.}$ is the row vector constituted by the element in the i-th row of matrix N, and $\boldsymbol{M}_{j.}$ corresponds to the column vector constituted by the element in the j-th row of matrix $\boldsymbol{M}$. NMI measures the similarity between the detection results and the real results. The higher the similarity, the closer the NMI value is to 1.

Query setting: For each dataset, we collect a set of meta-paths reported in Table 2. Note that in line with existing works[19,20], we focus on meta-paths with lengths at most four. We generate 100 queries for each dataset. To generate a query, we randomly select a meta-path and then select vertex. The results reported in the following, each data point is the average result for these 100 queries. When comparing the methods in this paper, Basic-core simply follows the best parameter settings from the original paper.

5.2 Performance Comparison (RQ1)

5.2.1 Parameter analysis

In this subsection, to avoid bias due to the different setting of parameters used during the random walk, we vary parameters of CSQR in each experiment to study the effect of parameters on the community search performance.

Our method includes three important parameters: α,β and K, where α controls the proportion of historical interaction information in the random walk process, β controls the ratio between previous visiting history and the average of key position vectors in the current time window in the process of aggregating historical information, K is the size of the current time window.

In Fig.4 the effect of α is studied. The value of α is varied on all datasets to investigate its influence on CSQR. The experimental result is shown in Fig.4. We can observe that the trend over all data sets is basically the same. As α increases, the performance shows a decreasing trend. The best performance is reached when α is taken as 0.2. Obviously, a smaller α value gives better performance. This is also straightforward since a smaller α value indicates that visiting history plays a more important role in deciding next steps of the walker. Although historical interaction information is important in deciding the next steps of the walker, information about jumps between nodes also needs to be taken into account, as this is an intrinsic feature of random walks.

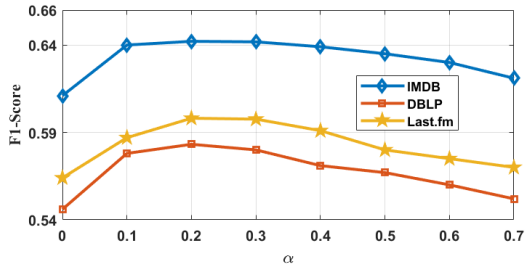


Fig.4 The effects of α

In Fig.5 the effect of β on our method is shown on all datasets. We can observe that when $0.3 \leqslant \beta \leqslant 0.6$, CSQR achieves the best performance. When $\beta \geqslant 0.8$, the performance drops. The reason is that when β is large, more faraway nodes from the query node will be assigned as key positions. The detection results include more false positive nodes. It obtains mediate F1-score performance when $\beta \leqslant 0.4$. Because insufficient true key positions can be covered.

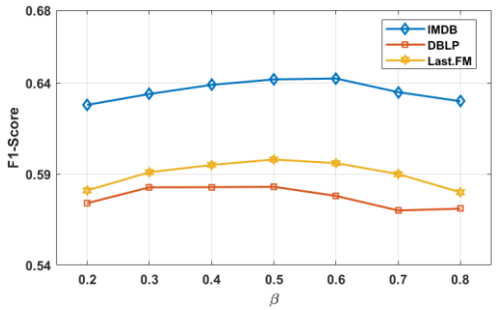


Fig.5 The effects of β

Fig.6 shows the effect of different values of K on our model for different datasets. We can observe that the trend is basically the same in all data sets. With increasing K, the F1-score also increases. This is consistent with our intuition, which is that larger K means that the current sliding window memorizes more previous steps.

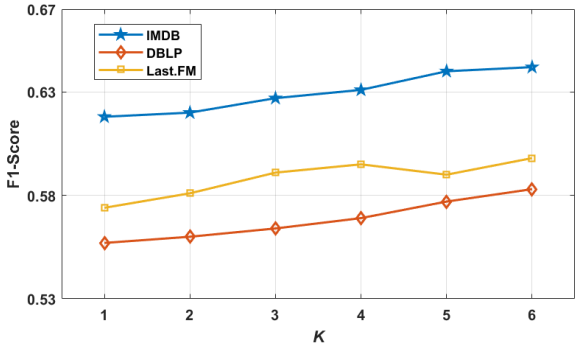


Fig. 6 The effects of *K*

5.2.2 Performance comparison

In this section, we compare our model with the existing baseline and various variants of ours. Towards this end, we perform experiments by using evaluation metrics F1-score and NMI.

To be fair, we experiment 100 times and reported the average results of the performance comparison on the three datasets in the Table 3. From the result, We summarize several important observations:

Table 3: Comparisons of overall performance between CSQR and baselines

Datesets	Metrics	Basic-core	CSQR-W	CSQR-R	CSQR-WR	CSQR
IMDB	*F1-score*	0.605	0.626	0.631	0.613	**0.642**
	NMI	0.594	0.611	0.629	0.603	**0.639**
DBLP	*F1-score*	0.542	0.561	0.572	0.559	**0.583**
	NMI	0.535	0.553	0.566	0.547	**0.579**
Last.fm	*F1-score*	0.563	0.573	0.584	0.554	**0.598**
	NMI	0.552	0.564	0.571	0.546	**0.587**

(1) Basic-core achieves poor performance on three datasets. This implies that strict cohesion metrics are difficult to adapt to real-world community structures. Also, Basic-core fails to take into account the intermediate nodes of meta-paths, as the homogeneous graphs induced directly based on the neighbors of meta-paths are insufficient to retain richer semantic information, which limits the performance of the model.

(2) CSQR-W achieves better performance than BASIC on three datasets. This further proves that the community structure is difficult to satisfy the strict cohesiveness metric, and although both omit thef intermediate nodes of the meta-path, CSQR-W obtains the community structure related to the query node by the random walk strategy, which is more flexible and better adapted to the community structure compared to Basic-core.

(3) Compared to CSQR-R, the performance of CSQR verifies that query node quality is extremely significant in community search tasks. Since the quality of user-defined seed node is uncertain, the poor quality of seed node can seriously influence the result of random walk. Therefore, we are more interested in node that have a high clustering tendency and is similar to the query node.

5.3 Community quality analysis (RQ2)

To further analyze the performance of our method, in this section, following [5], we analyze the quality of the community in the perspective of Density and Similarity.

Density of link relationships. Conventionally, the density of a graph is defined as the number of edges over the number of vertices [16,21]. To adapt it for communities in HINs, Fang[5] et al redefine it as the number of vertex pairs that are $\mathcal{P}$-connected over the number of vertices (here, all the vertices are with the target type). Fig.7 demonstrates the average density of communities found by different methods on all networks, and we observe that the CSQR-based communities have the highest average density.

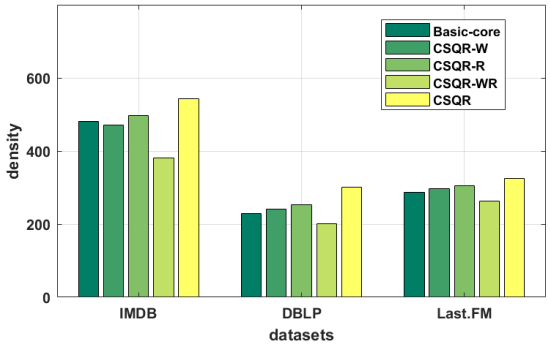


Fig. 7 The average density of community

Similarity of community members. We have measured the similarity of community members by PathSim[20]. Specifically, we compute the PathSim value for each pair of vertices in the communitiy.Fig.8 shows the average PathSim values on three datasets. Clearly, communities of CSQR achieve higher similarity values than those of other, so their members are more similar to each other.

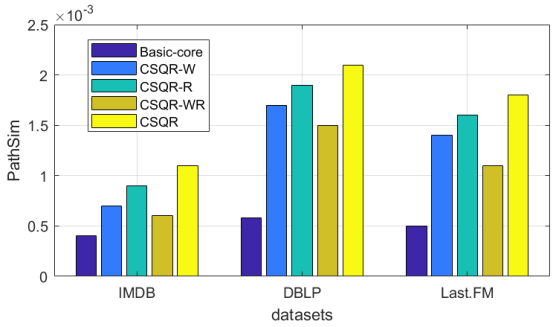


Fig.8 The similarity of community members

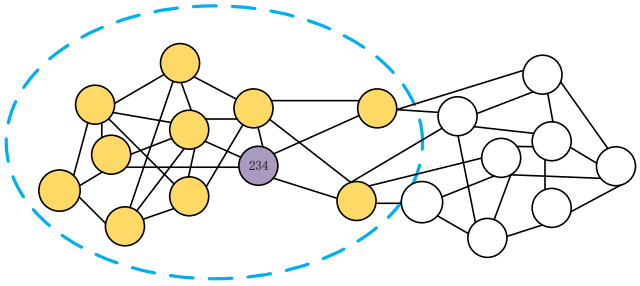
5.4 Component contribution analysis (RQ3)

In this subsection, we answer the third question, i.e., how much do the two main components of CSQR, induced weighted homogeneous graph and query node

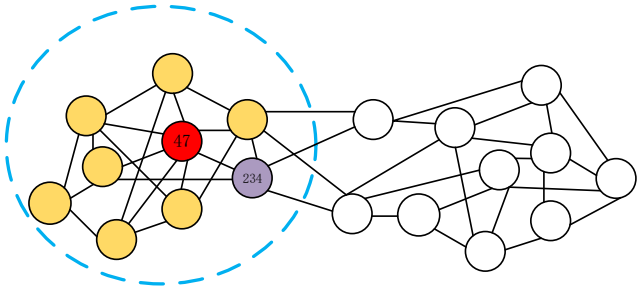
replacement, contribute to the model? Three variations of CSQR are discussed for this study. Their performances are also reported in Table 3. From the result, we can get the following conclusions:

It can be observed that average F1-score and NMI score of CSQR-W ignoring information about intermediate nodes of the meta-path show a lower value on each dataset than that of the CSQR method, which indicates the effectiveness of the information about intermediate nodes of the meta-path. Meta-paths are beneficial tools for capturing the rich semantic information of heterogeneous information networks, and ignoring the information of intermediate nodes can result in a loss of information, which is very important for community search tasks on heterogeneous networks.

In the same way, we can find a significant performance decrease of CSQR-R compared to CSQR. This indicates that the query node replacement strategy is very critical for our model. The query replacement strategy can not only select high-quality nodes to prepare for random walk, but also improve the accuracy of the community search algorithm. In a word, the query replacement strategy is indispensable in our model.



(a) CSQR-R



(b) CSQR

Fig.9 Experimental result of CSQR-R and CSQR in DBLP dataset

In order to be able to sufficiently explain that seed replacement is effective for solving the query bias problem, Fig.9 shows the boundary part of the experimental results for CSQR-R and CSQR on the DBLP dataset. Node 234 is the given initial query node, and the subgraph consisting of the orange nodes indicates the community located by the corresponding method. The experimental result for CSQR is given in Fig.9(a), suggesting that the returned community contains some noisy nodes. In contrast, Fig.9(b) indicates that query node 234 is replaced with core node 47. Intuitively, node 47 has a better quality compared to node 234 and therefore searches for a more reasonable community.

6 Conclusions and Future Work

This paper studies the problem of community search, which aims to search a community for a query vertex in an HIN. Specifically, we design a weighting strategy and query node replacement for the community search problem on heterogeneous networks. The weighting strategy enables the induced homogeneous graph to contain more semantic information; the query node replacement strategy can find better quality nodes for community search. We conduct extensive experiments on real-world graphs to show that our proposed methods can provide high-quality results over HINs.

Currently, we focus on community composed of the same type of nodes. Therefore, in the future, we will study how to search community with vertices of multiple types (e.g., a community contains both authors and topics in the DBLP network) and community search given multiple query nodes.

Acknowledgements This work is supported by the National Natural Science Foundation of China (61762078, 61363058, 61966004),Natural Science Foundation of Gansu(21JR7RA114), Northwest Normal University Young Teachers Research Capacity Promotion Plan (NWNU-LKQN2019-2).

References

1. Bunimovich L A, Wang C J, Chae S, Webb B Z. Uncovering hierarchical structure in social networks using isospectral reductions. In: Proceedings of the

International Conference on Advances in Social Networks Analysis and Mining. 2018, 1199-1206
2. Ni J C, Chang S Y, Liu X, Cheng W, Chen H F, Xu D K, Zhang X. Co-regularized deep multi-network embedding. In: Proceedings of the 2018 World Wide Web Conference. 2018, 469-478
3. Van Essen D C, Smith S M, Barch D M, Behrens T E J, Yacoub E, Ugurbil K. The WU-Minn human connectome project: an overview. Neuroimage, 2013, 80:62-79
4. Fang, Y X., Huang, X., Qin L, Zhang Y, Zhang W J, Cheng R, Lin X M. A survey of community search over big graphs. The VLDB Journal, 2020, 29(1):353–392
5. Fang Y X, Yang Y X, Zhang W J, Lin X M, Cao X. Effective and efficient community search over large heterogeneous information networks. The VLDB Endowment, 2020, 13(6):854-867
6. Tong H H, Faloutsos C, Pan J Y. Fast random walk with restart and its applications. In: Proceedings of the 6th IEEE International Conference on Data Mining. 2006, 613-622
7. Shan J, Shen D R, Nie T Z, Kou Y, Yu G. Searching overlapping communities for group query. World Wide Web, 2016, 19(6):1179-1202
8. Akbas E, Zhao P X. Truss-based community search: a truss-equivalence based indexing approach. The VLDB Endowment, 2017, 10(11):1298-1309
9. Huang X, Lakshmanan L V S, Xu J L. Community search over big graphs. Synthesis Lectures on Data Management, 2019, 14(6):1-206
10. Sozio M, Gionis A. The community-search problem and how to plan a successful cocktail part. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2010, 939-948
11. Huang X, Lakshmanan L V S, Yu J X, Cheng H. Approximate closest community search in networks. The VLDB Endowment, 2015, 9(4):276–287
12. Huang X, Cheng H, Qin L, Tian W T, Yu J X. Querying k-truss community in large and dynamic graphs. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. 2014, 1311-1322
13. Andersen R, Chung F R K, Lang K J. Local graph partitioning using pagerank vectors. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science. 2006, 475-486
14. Jian X, Wang Y, Chen L. Effective and efficient relational community detection and search in large dynamic heterogeneous information networks. The VLDB Endowment, 2020, 13(10):1723-1736
15. Bian Y C, Yan Y W, Cheng W, Wang W, Luo D S, Zhang X. On multi-query local community detection. In: Proceedings of the 18th IEEE International Conference on Data Mining. 2018, 9-18
16. Yang J, Leskovec J. Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems, 2015, 42(1):181-213
17. Fortunato S. Community detection in graphs. Physics reports, 2010, 486(3-5):75-174
18. Cantador I, Brusilovsky P, Kuflik T. Second workshop on information heterogeneity and fusion in recommender systems. In: Proceedings of the 2011 ACM Conference on Recommender Systems. 2011, 387-388
19. Huang Z P, Zheng Y D, Cheng R, Sun Y Z, Mamoulis N, Li X. Meta structure: Computing relevance in large heterogeneous information networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, 1595-1604
20. Sun Y Z, Han J W, Yan X F, Yu P S, Wu T Y. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. The VLDB Endowment, 2011, 4(11):992-1003
21. Wu Y B, Jin R, Li J, Zhang X. Robust local community detection: on free rider effect and its elimination. The VLDB Endowment, 2015, 8(7):798-809
22. Bian Y C, Huan J, Dou D J, Zhang X. Rethinking Local Community Detection: Query Nodes Replacement. In: Proceedings of the 20th IEEE International Conference on Data Mining. 2020, 930-935