K-Hop Community Search Based on Local Distance Dynamics

Lijun Cai[1], Tao Meng[1(✉)], Tingqin He[1], Lei Chen[1], and Ziyun Deng[2]

[1] College of Information Science and Engineering, Hunan University, Changsha 410082, China
{ljcai,mengtao,hetingqin,chenleixyzl23}@hnu.edu.cn
[2] Changsha Commerce and Tourism College, Changsha 410082, China
dengziyun@l26.com

Abstract. Community search aims at finding a meaningful community that contains the query node and also maximizes (minimizes) a goodness metric, which has attracted a lot of attention in recent years. However, most of existing metric-based algorithms either tend to include the irrelevant subgraphs in the identified community or have computational bottleneck. Contrary to the user-defined metric algorithm, how can we search the natural community that the query node belongs to? In this paper, we propose a novel community search algorithm based on the concept of k-hop and local distance dynamics model, which can natural capture a community that contains the query node. Extensive experiments on large real-world networks with ground-truth demonstrate the effectiveness and efficiency of our community search algorithm and has good performance compared to state-of-the-art algorithm.

Keywords: Community search · Interaction model · Complex network

1 Introduction

Most complex networks in nature and human society, such as social networks and communication networks, contain community structures. The goal of community detection is to identify all communities in the entire network, which is a fundamental graph mining task which has been well-studied in the literature [1–3]. Recently, a different but related problem called community search have studied, which is to find the most likely community that contains the query node [4]. It has a wide range of applications in complex networks analysis, such as social contagion modeling and social circle detection [5].

In all the previous studies on these problems, a goodness metric is usually used to identify whether a subgraph forms a community. Many approaches have been proposed to find a subgraph contains the query node and the goodness metric is maximized or minimized, such as k-core [6, 7], k-truss [8, 9] and densest graph [10]. However, most of the existing goodness metrics do not address the "free rider effect" issue, that is, nodes irrelevant to query node or far away from it are included in the identified community [9, 10]. Moreover, real-world applications often generate massive-scale graphs and require efficient processing. Therefore, achieving strong scalability together with high-quality community search is still a challenging, open research problem.

D. Liu et al. (Eds.): ICONIP 2017, Part V, LNCS 10638, pp. 24–34, 2017.
https://doi.org/10.1007/978-3-319-70139-4_3

In this paper, instead of introducing a new goodness metric for community search like k-core or k-truss, we consider the problem of community search from a new point view: local distance dynamics. The basic idea is to envision the nodes which k-hop away from a query node as an adaptive local dynamical system, where each node only interacts its local topological structure. Relying on a proposed local distance dynamics model, the distances among node will change over time, where the nodes sharing the same community with the query node tend to gradually move together while other nodes will keep far away from each other. Such interplay eventually leads to a steady distribution of distances and a meaningful community is naturally found.

The remainder of the paper is organization as follow: Sect. 2 gives related preliminary with our work. The details of our proposed algorithm are described in Sect. 3. Extensive experimental evaluation is presented in the Sect. 4. Section 5 provides our brief conclusion.

2 Preliminary

For the purpose of community search, some necessary definitions are first introduced. In this paper, we focus on an undirected and unweighted simple graph $G = (V, E)$, where V and E are a set of nodes and edges, respectively. Other type of graphs, such as directed and weighted can be handled with only slight modifications.

The structure of a node can be described its neighbors, and the distance between two nodes always according to how they share neighbors. The neighbors of a node is a node set of composed of all its adjacent nodes and the node itself.

Definition 1 (neighbors of node u). Given an undirected graph $G = (V, E)$, the neighbors of node u, denoted by $N(u)$, and is defined as follows:

$$N(u) = \{v \in V | \{u, v\} \in E\} \cup \{u\} \tag{1}$$

In order to discover local community for a given node, the method first need to initialize the distance for each edge. We use the popular Jaccard Distance [5] to measure the initial distance between two adjacent nodes.

Definition 2 (Jaccard Distance). Given an undirected graph $G = (V, E)$, the Jaccard Distance between node u and node v is defined as:

$$d(u, v) = 1 - \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \tag{2}$$

3 Our Algorithm

3.1 Problem Definition

In order to efficiently identify the scope of the community that the query node belongs to, we use an observation of real-world graphs: the best community for a given node is

in the neighborhood of the node. This observation is based on a well-known property of real-world graphs: small world effect, which is the name given to the finding that the average path (hop) between vertices in a network is small, and the local structure of network still has obvious grouping characteristics [11, 12].

Based on this property, we can prunes the distance evaluation for the nodes that are more than k-hop away from the query node. Specifically, our algorithm first roughly detects a k-hop subgraph by searching the nodes are k-hop away from the query node. It then refines the k-hop subgraph to find a best community that the query node belongs to by the local distance dynamics model. Before proceeding further, we given the formal definition of k-hop subgraph is as follows.

Definition 3 (K-Hop Subgraph). Given a graph $G = (V, E)$, a query node $q \in V$ and an integer $k > 0$. G' is a k-hop subgraph if and only if G' is connected, and each node u in G' has distance at most k-hop away from the node q.

On the basis of the definitions of k-hop subgraph, we give the definition of the k-hop community as follows, where the parameter k controls the scope of the community.

Definition 4 (K-Hop Community). Given a graph $G = (V, E)$, a query node $q \in V$ and an integer $k > 0$. C is a k-hop community, if C satisfies the following constrains.

- **Connectivity.** C is connected k-hop subgraph and contained node q.
- **Cohesiveness.** Employing the proposed local distance dynamics model on k-hop subgraph, where nodes in the target community will move together while other nodes will keep far away from the query node q.

Clearly, the *connectivity* constrain requires that the k-hop community containing the query node q be connected. In addition, the *cohesiveness* constrain makes sure that each node is as close as possible to the query node in the k-hop community. With the connectivity and cohesiveness constrains, we can ensure that the k-hop community is a connected and cohesive subgraph. The following example illustrate the definition of k-hop community.

Let us consider the graph show in Fig. 1. Assume that $k = 1$ and q is the query node. By Definition 3, we can see that the 1-hop subgraph included by node set $\{q, h_{11}, h_{12}, h_{13}, h_{14}, h_{15}, h_{16}\}$. However, it includes node h_{13} which is intuitively not relevant

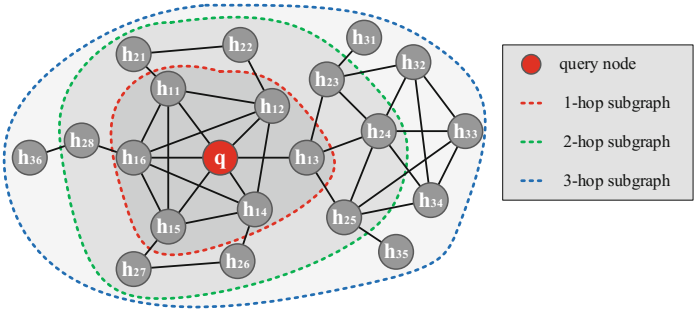

Fig. 1. An example graph for k-hop community.

to the query node. Relying on a proposed local distance dynamics model, the nodes q, h_{11}, h_{12}, h_{14}, h_{15} and h_{16} are moving together, while the node h_{13} keeping far away from them. As a result, the target community for node q is $C = \{q, h_{11}, h_{12}, h_{14}, h_{15}, h_{16}\}$.

Problem Definition. The problem of k-hop community search studied in this paper is defined as follows. Given a graph $G = (V, E)$, a query node $q \in V$ and an integer $k > 0$, find a k-hop community containing q.

3.2 Local Distance Dynamics Model

After specifying the scope of the target community, the next crucial step is to determine the interaction model among nodes in k-hop subgraph to simulate the distance dynamics. In the following, we will elaborate how the distance changes in local distance dynamics model.

Definition 5 (Core Edge). Given a k-hop subgraph $G'(V', E') \subseteq G(V, E)$, the edge $e = \{u, v\} \in E$ is core edge *iff* $e = \{u, v\} \in E'$.

As shown in Fig. 2(a). In this example network, node $q \in V$ is the query node and $k = 1$, and the circled by dotted line denote 1-hop subgraph. According to Definition 5, the edges *e(u,v)*, *e(u,a)*, *e(u,b)*, *e(u,q)*, *e(v,a)*, *e(v,b)*, *e(v,q)*, *e(a,b)*, *e(a,q)* and *e(b,q)* are core edges since them contained in 1-hop subgraph.

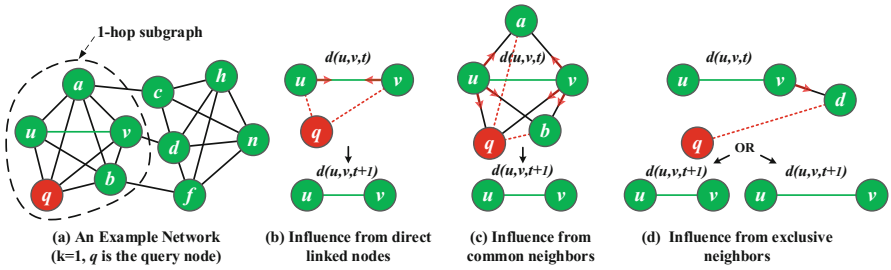

Fig. 2. Distance dynamics of one core edge influencing by three distinct interaction patterns.

Formally, let $e = \{u, v\} \in E$ be a core edge between two adjacent nodes u and v, and the $d(u,v)$ is its initial distance. Obviously, any change of the distance $d(u,v)$ actually results from the variation of node u and node v. Relying on its complete local topological structure (see Fig. 2), there are three distinct interaction patterns that allows influencing the distance $d(u,v)$.

Local Pattern 1. Here we consider the first interaction pattern: influence from direct linked nodes u and v (see Fig. 2(b)). Through mutual interactions, one node attracts another to move towards itself, and thus leads to the decrease of distance $d(u,v)$. Formally, we define the change of $d(u,v)$ from the influence of the direct linked nodes, *DI*, as follows:

$$DI = -\left(\frac{f(1-d(u,v)).(1-d(u,q))}{deg(u)} + \frac{f(1-d(u,v)).(1-d(v,q))}{deg(v)}\right) \quad (3)$$

In pattern *DI*, *f(·)* is a coupling function and *f(.) = sin(.)* is used in this study. The term *1−d(.,.)* implies the similarity between two direct linked nodes *u* and *v*, the more similar the two node have, the higher influence they will have. The term *1/deg(.)* is a normalized factor which is used to consider the different influences between linked nodes with diverse degrees.

Take friendship network as an example. In general, each people affects their know people, and tends to increase their cohesiveness gradually. Moreover, the more similar the two people are, the higher influence between each other they will have; the more similar to the query people, the more likely to share the same community with the query people; the people with more friends are harder to be influenced comparing to the people with less friends.

Local Pattern 2. The second interaction pattern happens when there exists some common neighbors between nodes *u* and *v* (see Fig. 2(c)). The common neighbors between node *u* and *v*, denoted by *CN = (N(u)-u)∩(N(v)-v)*. As the common neighbors have both links with node *u* and *v*, they attract the two nodes to move towards itself, and thus result in the decrease of distance *d(u,v)*. Formally, to characterize the change of the distance *d(u,v)*, we define the *CI*, indicating the influence from the interactions of common neighbors, as follows:

$$CI = -\sum_{x \in CN}\left(\begin{array}{l}\frac{1}{deg(u)} \cdot f(1-d(x,u)) \cdot (1-d(x,v)) \\ \qquad + \frac{1}{deg(v)} \cdot f(1-d(x,v)) \cdot (1-d(x,u))\end{array}\right).(1-d(x,q)) \quad (4)$$

In pattern *CI*, the two terms *1 − d(x,u)* and *1 − d(x,v)* indicate the similarity of common neighbor *x* with node *u* and *v*, respectively. If the *x* is more similar to *u*, the influence from *x* on *v* is more similar to the influence from *u*. The term *1 − d(x,q)* implies the similarity between common neighbor *x* and the query node *q*, the more similar to the query node, the higher influence the common node *x* will have.

Let us reconsider the friendship network as an example. Obviously, when two people share many common friends, their similar degree becomes large, and tends to increase their cohesiveness gradually. Furthermore, if their common friends are close to the query people, they tend to share the same community with the query people.

Local Pattern 3. The influence from exclusive neighbors is the third interaction pattern (see Fig. 2(d)). The exclusive neighbors only belongs to node *u* or *v*, and donated by *EN(u) = N(u) − (N(u)∩N(v))* and *EN(v) = N(v) − (N(u)∩N(v))*, respectively. In this pattern, each exclusive neighbor may have the positive or negative influence to the distance *d(u,v)*. To determine the positive or negative influence of exclusive neighbors on the distance, a similarity-based heuristic strategy is proposed. The basic idea is to investigate whether each exclusive neighbor of node *u* is similar

with the query node q, and vice versa. If the exclusive neighbor of node u is similar with the query node q, the movement of node u towards the exclusive neighbor results in the decrease of distance $d(u,v)$. Formally, we define the degree of positive or negative influence on the distance $d(u,v)$ from the exclusive neighbor as follows:

$$\sigma(x,q) = \begin{cases} (1-d(x,q)) & (1-d(x,q)) \geq \lambda \\ (1-d(x,q)) - \lambda & \text{otherwise} \end{cases} \tag{5}$$

In the above Eq. 5, the term λ is a cohesive parameter, and will be further discussed in Sect. 4.2. Then, we define the change of $d(u,v)$ from the influence of exclusive neighbors, EI, as follows:

$$EI = \begin{pmatrix} -\sum\limits_{x \in EN(u)} \left(\frac{1}{deg(u)} \cdot f(1-d(x,u)) \cdot \sigma(x,q) \right) \\ -\sum\limits_{y \in EN(v)} \left(\frac{1}{deg(v)} \cdot f(1-d(y,v)) \cdot \sigma(y,q) \right) \end{pmatrix} \tag{6}$$

Finally, by considering three interaction patterns together, the dynamics of the distance $d(u,v)$ on core edge $e(u, v)$ over time is govern by:

$$d(u,v,t+1) = d(u,v,t) + DI(t) + CI(t) + EI(t) \tag{7}$$

In the above Eq. 7, the term $d(u,v,t + 1)$ is the new distance at time step $t + 1$. $DI(t)$, $CI(t)$ and $EI(t)$ are three different influence from the directed nodes, common neighbors, and exclusive neighbors on the distance $d(u,v,t)$ at time step t.

4 Experiments

4.1 Experiments Setup

In our experiments, we compare our algorithm with two representative community search algorithms: *K-Core* [6] and *K-Truss* [8]. We implemented all algorithms in Python and ran the experiments on a Windows Server with 2 Intel Xeon E5-2600 series processors and 176 GB main memory. For all experiments, without further statement, *K-Core* and *K-Truss* specify the default value of k to 6.

We used six large real-world networks in our experiments: *Amazon*, *DBLP*, *Youtube*, *LiveJournal*, *Orkut* and *Friendster*. These networks are provided with ground-truth community memberships and publicly available at https://snap.stanford.edu/data. We test the performance of the three algorithms to search local community by *Relative Density, Diameter* and *F-socre*, which are widely adopted by other community search methods [6, 8, 10].

4.2 Influence of Parameters

Our *K-Hop* algorithm uses two parameters: k and λ. In this subsection, we investigate the influence of k and λ on result of community. We are interested in the changes of community size and the accuracies with the different value of k and λ.

We first give our analysis about the influence of k on community size and accuracies. For *K-Hop* algorithm, the parameter k is used to determine the scope of local interaction. In general, it is expect that the community size is small changes with k. To verify this conjecture, we studied the sensitive of parameter k in a LFR network. Similar results were obtained on other networks.

The results are show in Fig. 3 and they verify our conjecture. From the Fig. 3(a), we can clearly see that the community size is very small when k = 1. When k = 2 and larger, the community size is almost stable. From the Fig. 3(b), we can clearly see that k = 1 is the critical point on which the minimum F-score is found. After this, the F-score is almost stable.

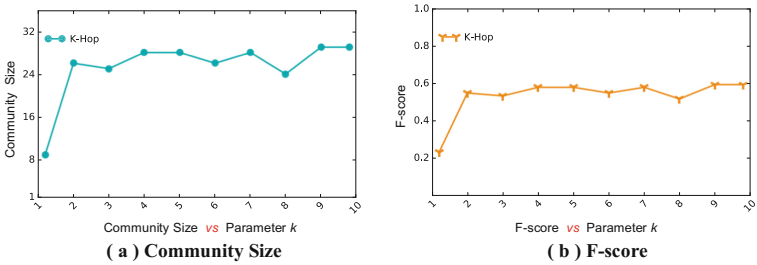

Fig. 3. Sensitive of parameter k.

The value of λ is the most important parameter for local distance dynamics model. For the *K-Hop* algorithm, the λ is used to determine the negative or positive interaction influence on the distances from exclusive neighbors. In general, it is expected that the community size monotonically decreases with λ.

Figure 4 shows our results on a synthetic network. From Fig. 4(a), we can clearly see that the community size monotonically decreases with λ. From the Fig. 4(b), we can clearly see that λ = 0.3 is critical point on which the maximal value of F-score is found. Before this, most of nodes in k-hop subgraph are move together to the query

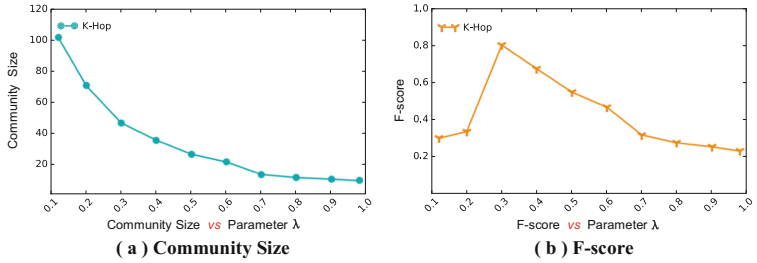

Fig. 4. Sensitive of parameter λ.

node when λ increase. After this, irrelevant nodes quickly keep far away from the query node due to the strong constraint on the closeness of a community.

From the results, we found that *K-Hop* is not sensitive to the searching results. In general, a λ value between 0.3 and 0.6 is normally sufficient to achieve a good result. We recommend a value for k, of 2. For *K-Hop*, we set the k = 2 and λ = 0.5 as default parameters.

4.3 Evaluation on Real Networks

We first evaluate the effectiveness of the selected methods on real networks. For each networks, we randomly select 100 query nodes with degree ranging from 10 to 100. The query node is selected from a random ground-truth community.

The Fig. 5(a) shows the relative density of the selected algorithms on different networks. It can be see that the *K-Truss* method better than other methods on most networks. Focus on the *K-Core* and *K-Hop* algorithms, it is not difficult to find that, the performance of *K-Hop* algorithm is exceeded to the *K-Core* method. In addition, the *K-Hop* algorithm is very close to the *K-Truss* algorithm on some real-world networks.

Figure 5(b) discusses the diameter of community search of various algorithms on real-world networks. From the Fig. 5(b), we can get the following observations. (1) For *K-Core*, the value of the diameters on six networks are very uneven, which imply the performance of the *K-Core* algorithm is very unstable on real-world networks. (2) For *K-Truss*, we can find that, *K-Truss* has better result and stability than *K-Core* algorithm on real-world networks. (3) For *K-Hop*, six real-world networks have the good results, the average value of the diameter is lesser than 4. (4) Focus on *K-Core* and *K-Truss* two algorithms, we can observe that these algorithms have larger diameter, this may be caused by the free rider effect.

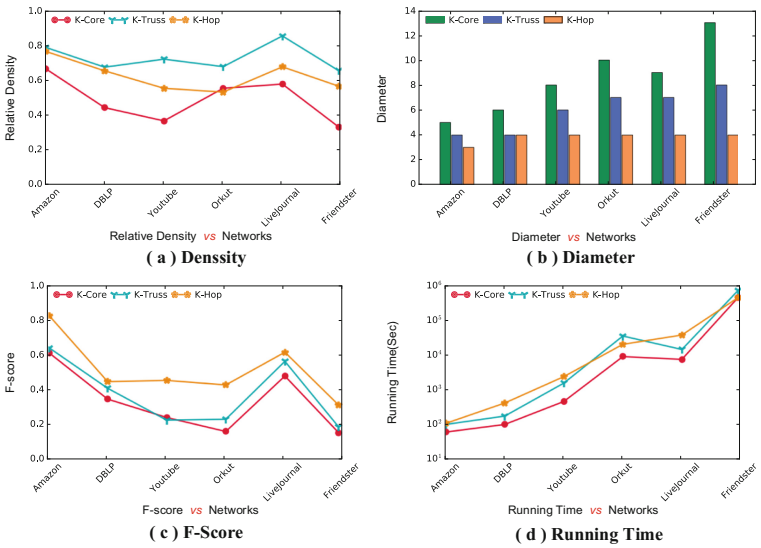

Fig. 5. Performance of community search of different algorithms on real-world networks.

Figure 5(c) shows the F-score of the identified community using different algorithms. We can see that the F-score value of *K-Hop* is 5% to 10% higher than those of other methods. If the nodes in the irrelevant community are selected as the identified community, these algorithms will identified irrelevant communities and will causes the low F-score value.

From the Fig. 5(d), when the scale of the network is small, the running time of all algorithms are small; Along with the increase of the scale of network, the running time are increase gradually. It is important to note that the *K-Core* is run faster than *K-Truss* and *K-Hop*, but it accuracies is low. The *K-Truss* and *K-Hop* algorithms have similar performance.

4.4 Case Study

To validate the effectiveness of our local distance dynamics model, we select two well-known UCI real-world networks with ground truth, use *K-Hop* algorithm to search the community structure with different query node.

The first network is the Zachary's karate club network, consisting of 34 vertices and 78 undirected edges. In this case study, we use node "1" and "34" as the query node, respectively. After set k = 2 and λ = 0.5, we got the community result shown in Fig. 6. Figure 6(a) shows the ground truth of karate club network, which covers 2 classes. Figure 6(b) shows the detection results with "1" as the query node, denoted by green nodes. Figure 6(c) shows the detection results with "34" as the query node, denoted by green nodes.

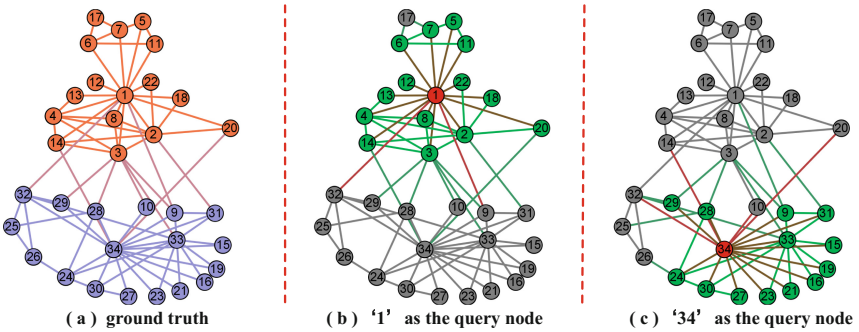

Fig. 6. Case study on the Zachary's karate club. (Color figure online)

The second network is Books about US politics network, consisting of 105 nodes and 441 edges. Here, we use node "8" and "66" as the query node, respectively. After set k = 2 and λ = 0.5, we got the community result shown in Fig. 7. Figure 7(a) shows the ground truth of network, covering 3 classes. Figure 7(b) shows the detection results with "8" as the query node, denoted by green nodes. Figure 7(c) shows the detection results with "66" as the query node, denoted by green nodes.

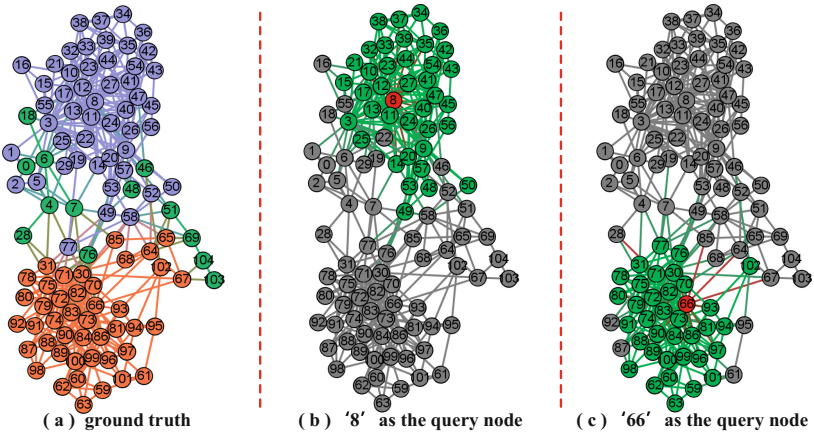

Fig. 7. Case study on the Books about US politics. (Color figure online)

5 Conclusions

In this paper, we introduce a new community search algorithm, called K-Hop, to automatically find the best community containing a query node in networks based on local distance dynamics. Extensive experimental is executed on six real-world networks show the effectiveness and efficiency of local distance dynamics model and search algorithm. Our future work will consider the community search on heterogeneous network based on the intuitive local distance dynamic model.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (61174140, 61472127, 61272395); China Postdoctoral Science Foundation (2013M540628, 2014T70767); Natural Science Foundation of Hunan Province (14JJ3107); Excellent Youth Scholars Project of Hunan Province (15B087).

References

1. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.: Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 824–833. ACM (2007)
2. Shao, J., Han, Z., Yang, Q., Zhou, T.: Community detection based on distance dynamics. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1075–1084. ACM (2015)
3. Newman, M.E.: Modularity and community structure in networks. Proc. Nat. Acad. Sci. **103**, 8577–8582 (2006)
4. Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 939–948. ACM (2010)
5. Ugander, J., Backstrom, L., Marlow, C., Kleinberg, J.: Structural diversity in social contagion. Proc. Nat. Acad. Sci. **109**, 5962–5966 (2012)

6. Cui, W., Xiao, Y., Wang, H., Wang, W.: Local search of communities in large graphs. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 991–1002. ACM (2014)
7. Li, R.H., Qin, L., Yu, J.X., Mao, R.: Influential community search in large networks. Proc. VLDB Endow. **8**, 509–520 (2015)
8. Huang, X., Cheng, H., Qin, L., Tian, W., 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, pp. 1311–1322. ACM (2014)
9. Huang, X., Lakshmanan, L.V., Yu, J.X., Cheng, H.: Approximate closest community search in networks. Proc. VLDB Endow. **9**, 276–287 (2015)
10. Wu, Y., Jin, R., Li, J., Zhang, X.: Robust local community detection: on free rider effect and its elimination. Proc. VLDB Endow. **8**, 798–809 (2015)
11. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393** (6684), 440–442 (1998)
12. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity – a proper metric. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 166–181. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23059-2_15