BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 02/06 pp171–180
Volume 3, Number 3, September 2020
DOI: 10.26599/BDMA.2020.9020002

Fast Skyline Community Search in Multi-Valued Networks

Dongxiao Yu, Lifang Zhang*, Qi Luo*, Xiuzhen Cheng, Jiguo Yu, and Zhipeng Cai

Abstract: Community search has been extensively studied in large networks, such as Protein-Protein Interaction (PPI) networks, citation graphs, and collaboration networks. However, in terms of widely existing multi-valued networks, where each node has d ($d \geqslant 1$) numerical attributes, almost all existing algorithms either completely ignore the attributes of node at all or only consider one attribute. To solve this problem, the concept of skyline community was presented, based on the concepts of k-core and skyline recently. The skyline community is defined as a maximal k-core that satisfies some influence constraints, which is very useful in depicting the communities that are not dominated by other communities in multi-valued networks. However, the algorithms proposed on skyline community search can only work in the special case that the nodes have different values on each attribute, and the computation complexity degrades exponentially as the number of attributes increases. In this work, we turn our attention to the general scenario where multiple nodes may have the same attribute value. Specifically, we first present an algorithm, called MICS, which can find all skyline communities in a multi-valued network. To improve computation efficiency, we then propose a dimension reduction based algorithm, called P-MICS, using the maximum entropy method. Our algorithm can significantly reduce the skyline community searching time, while is still able to find almost all cohesive skyline communities. Extensive experiments on real-world datasets demonstrate the efficiency and effectiveness of our algorithms.

Key words: multi-valued graph; community search; skyline community

1 Introduction

Community search is a fundamental problem in network analysis, which has attracted much attention recently due to its wide applications, such as protein structure analysis[1], organization of activities[2], and advertising[3]. The community search problem is a query-dependent community discovery problem, and it requires to find densely-connected subgraphs in a network given query conditions.

In real-world applications, there exists one important network called multi-valued network, where the nodes are associated with d ($d \geqslant 1$) numerical attributes. We can notice that in multi-valued networks, besides the structural cohesiveness constraint, the community search problem also needs to consider the influence of the community represented by attribute values of nodes. A challenging task is how to define a community, which is dominated by other communities in terms of d numerical attributes so that the most important communities can be found. To solve this problem, a novel community model, called the skyline community model, was presented, based on the concepts of k-core[4] and skyline[5, 6]. In particular, the skyline community is a maximal connected k-core that is not dominated by

• Dongxiao Yu, Lifang Zhang, Qi Luo, and Xiuzhen Cheng are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China. E-mail: dxyu@sdu.edu.cn; fzhanglf@mail.sdu.edu.cn; luoqi2018@mail.sdu.edu.cn; xzcheng@sdu.edu.cn.
• Jiguo Yu is with the School of Computer Science and Technology, Qilu University of Technology, Jinan 250353, China. E-mail: jiguoyu@sina.com.
• Zhipeng Cai is with Department of Computing Science, Georgia State University, Atlanta, GA 30303, USA. E-mail: zcai@gsu.edu.
∗To whom correspondence should be addressed.
Manuscript received: 2020-02-29; accepted: 2020-03-07

other connected k-cores in the d-dimensional attribute space. A space-partition algorithm to efficiently compute the skyline communities was also presented in Ref. [7]. However, the proposed algorithm only works in a very special setting that the values of nodes on each attribute are totally different. Obviously, this setting is unrealistic in real-world applications, as nodes frequently have the same value on an attribute. Hence, an open problem is then whether we can derive an algorithm for skyline community search in the general scenario. On the other hand, the efficiency of the space-partition algorithm degrades exponentially with the increase of the number of attributes, which hinders the application of the existing algorithm in scenarios of a large number of attributes.

To solve the above problems, we study the skyline community search in a general multi-valued network, where multiple nodes may have the same value on an attribute. We present an algorithm, called MICS, which can compute all skyline communities in general multi-valued networks. A Maximal Community Influence (MCI)-centric (see Section 3 for detailed definition) approach is adopted instead of the node-centric one in Ref. [7], to avoid the relying on different attribute values of nodes. Furthermore, due to the fact that in real-world applications, it usually only needs to search important communities, i.e., those with high cohesiveness, it admits to improve the computation efficiency by ignoring communities that are less important. With this intuition, we present an algorithm, called P-MISC, which can significantly reduce the computation time of the skyline community search. The algorithm adopts the maximum entropy method to discover a small number of crucial attributes, which can reserve the information of the network as much as possible. Consequently, this approach can help find almost all high cohesive skyline communities. Extensive experiments verify the efficiency and effectiveness of the proposed algorithms.

The rest of this paper is organized as follows. We review the related work in Section 2 and present some basic concepts and problem description in Section 3. Section 4 introduces the basic multidimensional algorithm MICS and the improved algorithm P-MICS after dimension reduction. Our experimental results are reported in Section 5. At last, the paper is concluded in Section 6.

2 Related Work

In this section, we review some existing related studies, mainly including Community Search (CS) and community search on the attribute graph.

Community search. Discovering communities structure from a large social network has been a very popular topic. To solve this problem, two basic concepts of CS[8] and Community Detection (CD)[9, 10] have been proposed. They are both designed to find some cohesive subgraphs called communities from a large network. The difference between them is that the solutions to CS are generally given some query nodes or query parameters, while CD will find all communities. Besides, compared with CD, CS consumes less time and is more suitable for the study of cohesive subgraphs in large graphs. Sozio and Gionis[2] proposed to find a connected k-core containing query nodes, and it is also believed to be the first algorithm that adopts a global approach. In Ref. [11], a community search algorithm based on the local expansion techniques was proposed, and it greatly improves the efficiency of the global algorithm. In addition to k-core[2, 11–17], there are many other classic indicators to measure the cohesiveness of the communities, such as k-truss[18, 19], k-plex[20], k-clique[21, 22], etc. However, most of the methods mentioned above are limited to the structural properties of the community and ignore the attributes of the nodes.

Community search on attribute graphs. In fact, community search on the multiple attribute graph has made great progress. In addition to the basic structural constraints, existing works utilize different definitions and methods in terms of attribute constraints. In Ref. [23], each node was associated with a set of keywords, and it tried to find the community with the maximum number of common keywords for all nodes. Besides, there are also some algorithms on a location-based attributed graph, such as Spatial-Aware Community (SAC) search[24], Radius-Bounded k-core (RB-k-core) search[25], and Geo-Social Group Queries with minimum acquaintance constraint (GSGQ)[26]. Recently, Li et al.[27] assigned each node a weight (such as PageRank or any other user-defined attributes) and formulated the influence of a community as the minimum weight among all nodes. In many real social networks, each node has a number of attributes that represent different aspects of the node. However, the algorithms discussed above take into account only one attribute. Based on the definition of influential community in Ref. [27], Li et al.[7] further studied the first community model for a multi-valued network, and the communities focusing on are those that cannot be dominated by other communities. However,

when searching all the skyline communities, the time complexity of the algorithm increases exponentially with the increase of the number of attributes, so it is difficult to handle networks with higher dimensions. Therefore, it is necessary to propose a more efficient way that can find the most important communities in less time.

3 Problem Definition

We first formally present the multi-valued graph model. The network with d numerical attributes is modeled as a multi-valued graph $G = (V, E, A)$, where V, E, and A represent the sets of nodes, edges, and d-dimensional vectors associated to nodes, respectively. We have $|V| = |A| = n$ and $|E| = m$. The d-dimensional real-valued vector associated to a node $u \in V$, which represents the attribute values of node u, is denoted as $A_u = (a_u^1, \ldots, a_u^d)$, where $A_u \in A$ and $a_u^i \in \mathbb{R}$ for $i = 1, 2, \ldots, d$. Without loss of generality, we assume that the attribute values are positive. Notice that different from Ref. [27], we do not assume that the values in a dimension form a strict order, which makes our result extensively applicable in reality.

We study the community search problem in the above defined multi-valued graph model. Intuitively, the problem is to find communities that have an influence on multiple dimensions. Specifically, we adopt the skyline community model that is presented in Ref. [7] to define the influential communities. The skyline community model is based on the concepts of k-core[4] and skyline[5], which aims to find communities that are not dominated by the other communities in terms of d numerical attributes. The k-core model well depicts the cohesiveness of a community, while the skyline model captures the influence of a community. We next introduce the definition of skyline community in detail.

Definition 1 (***k*-core**) Given an undirected graph $G = (V, E)$, and let $H = (V', E')$ is a subgraph of G, where $V' \subseteq V$ and $E' \subseteq E$. If H satisfies $\forall v \in H$, $\deg_H(v) \geqslant k$, then H is a k-core of graph G.

The k-core has been widely used to represent the structural cohesiveness of communities. However, the k-core model cannot capture the d-dimensional numerical attributes of a community. To better describe the importance of community, the skyline community model takes the numerical attributes into account.

Definition 2 (**Community Influence(CI)**) Given a multi-valued graph $G = (V, E, A)$, assume that the minimum attribute value of nodes in the i-th dimension (for $i = 1, 2, \ldots, d$) of G is denoted by $f_i(G)$, i.e., $f_i(G) = \min_{u \in V}(A_u^i)$, and the CI of G is defined by $f(G) = (f_1, f_2, \ldots, f_d)$.

The definition of CI is defined as the minimum attribute value of a community in all dimensions, which ensures the minimum influence of nodes in the target community. If the context is clear, we simply write $f_i(G)$ as f_i. With this definition, it is easy to compute the influential communities if there is only one dimension. The basic idea is that we can first compute the largest CI among the communities satisfying the cohesive property in a graph, denoted as CI*, and delete the nodes whose attribute values are smaller than CI*. The connected k-cores in the remaining graph are then the communities that satisfy both the cohesiveness and the influence requirement.

However, when considering multiple dimensions, the situation becomes much more complicated, as it is not easy to define the maximum CI. For example, if there are two communities whose CIs are $(1, 3, 4)$ and $(3, 2, 1)$, respectively, and we cannot simply say which one is larger. Because we mainly care about the communities that are cohesive and cannot be dominated by other communities. Hence, the following MCI is proposed to measure the influence of communities.

Definition 3 (**MCI**) Let H_1 be a subgraph of G. If there does not exist another subgraph H_2 of G satisfying $f_i(H_1) = f_i(H_2)$ for all $i = 1, \ldots, d$, and $f_j(H_1) < f_j(H_2)$ for a certain j, then $f(H_1)$ is called a maximal community influence.

Furthermore, if for two communities H_1 and H_2, $f_i(H_1) = f_i(H_2)$ for all $i = 1, \ldots, d$, and $f_j(H_1) < f_j(H_2)$ for a certain j, we say H_1 is dominated by H_2, denoted as $H_1 \preceq H_2$.

Based on the definitions of k-core and MCI, the skyline community can then be defined.

Definition 4 (Skyline community[7]) Given a multi-valued graph $G = (V, E, A)$ and an integer k, a skyline community with a parameter k is a subgraph $H = (V_H, E_H, A_H)$ of G, such that H satisfies the following properties:

- Cohesive property: H is a connected k-core;
- Skyline property: $f(H)$ is an MCI;
- Maximal property: There does not exist an induced subgraph H' such that (1) H' is a k-core, (2) H' contains H, and (3) $f(H')$ is an MCI.

We use the example in Fig. 1 to illustrate the skyline community. In Fig. 1, each node has three attributes. Considering the case of $k = 2$, we will find that the

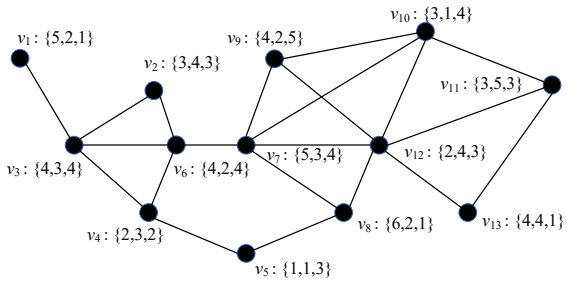


Fig. 1 Example of skyline communities.

subgraphs $H_1 = \{v_2, v_3, v_6\}$, $H_2 = \{v_7, v_9, v_{10}\}$, and $H_3 = \{v_{11}, v_{12}, v_{13}\}$ are skyline communities, as the CIs of these communities are $(3, 2, 3)$, $(3, 1, 4)$, and $(2, 4, 1)$, respectively, and there are no communities can dominate them. Notice that $H_5 = \{v_3, v_4, v_6\}$ is not a skyline community, since the CI of it is $(2, 2, 2) \preceq f(H_1)$.

With the definition of skyline community, the community search problem is then defined as finding all the skyline communities from a given multi-valued graph $G = (V, E, A)$ with the given parameter k. In Ref. [7], there have been algorithms proposed to solve the skyline community search problem. However, the complexity of these algorithms increases exponentially with the increase of the number of attributes, which makes the algorithms impossible to handle high-dimensional graphs. Hence, it deserves to investigat methods that can greatly improve the search efficiency, while can still list the most important communities, e.g., those with large parameter k. We propose a new approach that can reduce inessential attributes based on the concept of entropy defined below.

Maximum entropy model. The basic idea of our approach is to reduce inessential attributes such that the efficiency of community searching can be significantly improved. On the other hand, the searching result cannot be heavily affected after attribute reduction, or at least the important communities, such as those with large parameter k, should be found. Hence, the remaining attributes should keep as much information as possible that can facilitate finding the important communities. In principle, the maximum entropy theory states that, subject to precisely stated prior data (a set of precise constraints), the probability distribution which best represents the current state of knowledge is the one with the largest information-theoretical entropy. So we can use the entropy to help find the most important attributes.

Specifically, given a multi-dimensional graph $G = (V, E, A)$ and a parameter $p \leqslant d$, let A_p denote the set of all combinations of p dimensions. We calculate the set $P \in A_p$ consisting of the most important p dimensions, such that

$$\text{maximize} \quad -\sum_{a \in Q_P} \frac{N_P(a)}{N_P} \cdot \log\left(\frac{N_P(a)}{N_P}\right),$$
$$\text{subject to} \quad \forall_{a \in Q_P} N_P(a) > 0,$$

where Q_P denotes the set of values of nodes on the p dimensions in P; $N_P(a)$ denotes the number of nodes that have value a on the p dimensions in P, and N_P is the number of nodes that have values on the p dimensions in P, and in our setting, $N_P = n$ for any set P.

The following result ensures that we can use the approach of dimension reduction to improve the community searching efficiency.

Theorem 1 The communities we find after dimension reduction are skyline communities.

Proof Assuming that there are d dimensions, denoted by $A = \{1, 2, \ldots, d\}$. Without loss of generality, let $P = \{1, \ldots, p\}$ be the dimension selected. Let C be a skyline community with respect to the p dimensions in P, this means that there is no another skyline community C_1 that satisfies $f(C) \preceq f(C_1)$ on the dimensions in P. Thus, when considering all d dimensions, it is easy to see that $f(C)$ is still an MCI, and C clearly satisfies the other two properties. As a consequence, C is still a skyline community with d dimensions, and the theorem is proved. ■

4 Skyline Community Search Algorithm

In this section, we present our skyline community search algorithms. At first, we present the algorithm for the case of $d = 2$. The algorithm is similar to that in Ref. [7]. But we adapt the algorithm such that it can handle the scenario that nodes may have the same value on a dimension. Then with the above algorithm as a subroutine, we present the algorithm for the general case with a d-dimensional graph. We finally give a much more efficient algorithm with the idea of dimension reduction.

4.1 Algorithm for d=2

We consider the case that nodes have two attributes, denoted as (A^1, A^2). For a node u, let (a_u^1, a_u^2) denote the attribute values of u. For a community C, its CI is denoted as (f_1, f_2). Basically we need to figure out all (f_1, f_2) values that are MCI, based on which all skyline communities can be found.

Basically, we need to filter the k-cores to finally find skyline communities. The algorithm proposed in Ref. [7] relies on the assumption that the values of nodes on each particular dimension are all different, which admits that the algorithm just considers the k-cores containing some specific nodes. However, this idea does not work anymore when multiple nodes may have the same value on a particular dimension, as it cannot determine a unique node by a fixed value in a dimension. Hence, we propose a manner relying on MCIs to filtering k-cores.

Specifically, Algorithm 1 is first invoked to compute the largest CI on the A^2 dimension, denoted by f_2^*. Then, all nodes v with $a_v^2 < f_2^*$ are deleted from the graph. Notice that instead of actually deleting these nodes from the graph G, they are just removed temporarily so as to find out the communities whose CI on the A^2 dimension equals f_2^*. After that, Algorithm 1 is invoked again to find the largest value of nodes on the first dimension in the remaining graph, denoted as $\tilde{f}_1$. It is easy to check that $(\tilde{f}_1, f_2^*)$ is an MCI.

Once we obtain the MCI of $(\tilde{f}_1, f_2^*)$, we can immediately compute the corresponding communities. But how can we efficiently find other skyline communities? Notice that the value of f_2 in above MCI we have found is the largest on the A^2 dimension, so we have to increase f_1 to find other MCIs. Based on this idea, the nodes v with $a_v^1 \leqslant \tilde{f}_1$ can be deleted from G, as communities with these nodes are dominated by the community we have found. In the remaining graph, we iteratively use the same method to find the MCIs and corresponding skyline communities till $f_2 = 0$. The detailed algorithm is given in Algorithm 2. In Algorithm 2, dlist is used to store current (f_2, f_1) value. Since the algorithm computes f_2 first and f_1 later, we will also generate dlist in this order. Besides, fdf1 is a global variable that holds all the MCIs we have found. Initially, dlist and fdf1 are empty.

Theorem 2 In the case of $d = 2$, Algorithm 2 can compute all communities with time complexity of $O(l(m + n))$, where l is the maximum of different dimension values on every dimension.

Proof In Algorithm 2, MaxCI is first invoked to compute the maximum value on the A^2 dimension, which takes $O(n + m)$ time. After that, all MCIs and corresponding communities are iteratively calculated. The total number of iterations is upper bounded by l, and each iteration takes $O(n + m)$. Combining all together, the time complexity of the algorithm can be derived. ■

4.2 Algorithm for $d \geqslant 3$

In the scenario of $d = 3$, the method of computing the MCIs used in the case of $d = 2$, i.e., determining the largest value in one dimension and then going through the values on the other dimension, is unavailable any more. This is because in the case of $d \geqslant 3$, after computing the maximal value in one dimension, it is hard to determine the remaining values of a skyline community on other dimensions.

Our algorithm uses a similar dimension reduction idea

Algorithm 1 MaxCI (G, i, k)

Input: A multi-attribute graph $G = (V, E, A)$ and a non-negative integer k
Output: Largest CI in the i-th dimension

$f = 0$;
$G' \leftarrow$maximal k-core in G;
while $G' \neq \varnothing$ **do**
  Let $u \in G'$ be the node with the smallest attribute value on the i-th dimension;
  $f \leftarrow f_i(u)$;
  $G' \leftarrow$ Update(G', u);
return f;
Function Update(G, u);
foreach $v \in$ neighbor(u) **do**
  degree$(v) \leftarrow$ degree$(v) - 1$;
  if degree$(v) < k$ **then**
    Update(G, v);
delete u from G;
return G;

Algorithm 2 BICS (G, k, dlist)

Input: A multi-valued graph $G = (V, E, A)$ and a non-negative integer k
Output: Skyline communities of G

$R \leftarrow \varnothing, G' \leftarrow \varnothing$;
$G_1 \leftarrow$ the graph consisting of maximal k-cores in G;
$f_2^* \leftarrow$ MaxCI$(G, 2, k)$;
while $f_2^* > 0$ **do**
  $G' \leftarrow G_1$;
  remove $u \in G'$ with $a_u^2 < f_2$;
  $\tilde{f}_1 \leftarrow$ MaxCI$(G', 1, k)$;
  dlist $\leftarrow$ dlist $\cup \{f_2^*, \tilde{f}_1\}$;
  if dlist is not dominated by fdf1 **then**
    add dlist to fdf1;
    remove $u \in G'$ with $a_u^1 < \tilde{f}_1$;
    $G' \leftarrow$ kcore(G');
    $R \leftarrow$ connected communities of G';
  remove $u \in G_1$ with $a_u^1 \leqslant \tilde{f}_1$;
  $f_2^* \leftarrow$ MaxCI$(G_1, 2, k)$;
return R;

as that in the algorithm given in Ref. [7]. But here we need to implement the algorithm based on the MCIs, rather than some specific nodes. Basically, we first fix one dimension and derive all possible values on the dimension. Then, for each possible value on the fixed dimension, the algorithm invokes itself with dimensional parameter $d-1$ to compute the $(d-1)$-dimensional skyline communities. By checking whether the CI of these skyline communities is MCI, all d-dimensional skyline communities are finally found.

The detailed implementation of our algorithm is shown in Algorithm 3. Similar to Algorithm 2, the MaxCI method is first invoked to calculate the maximum f value of maximal k-cores on the d-th dimension, denoted by f^* (Line 6). Let F denote the set of all possible values on the d-th dimension. For each value $f \in F$ with $f \leqslant f^*$, the algorithm invokes itself to compute $(d-1)$-dimension skyline communities, after deleting all nodes with $a_v^d < f$ (Lines 7–14). When the dimension is reduced to 2, Algorithm 2 is invoked to compute 2-dimensional skylines (Lines 2 and 3). After determining the value of each dimension in the above procedure, it is then checked whether the CI composed by these values is maximal. Once an MCI is determined, its corresponding skyline communities are then found.

Theorem 3 For $d \geqslant 3$, Algorithm 3 can find all skyline communities, and the time complexity is $O(l^{d-2}(l(m+n)))$.

Proof For each CI generated in the algorithm, algorithm MICS will determine whether it is dominated by previous MCIs, and all possible combinations of $(f_1, ..., f_d)$ have been considered, so it is ensured that all MCIs have been found. According to each MCI, corresponding skyline communities are computed simultaneously. Hence, all skyline communities are finally found. We next discuss the time complexity of algorithm MICS.

In each dimension reduction process, the algorithm with a smaller dimension parameter is invoked for at most l times, and in general $l \ll n$. When d is reduced to 2, Algorithm 2 is invoked, which takes $O(l(m+n))$ time. Hence, the total time consumed is upper bounded by $O(l^{d-2}(l(m+n)))$. ■

As shown in Theorem 3, though Algorithm 3 can perfectly solve the skyline community search problem, its time complexity is unacceptable, which increases exponentially with the increase of the number of dimensions. Hence, we subsequently present an algorithm that can compute important skyline communities, while significantly reduce the time complexity, by considering only a fixed number of dimensions when computing the skyline communities.

4.3 Algorithm with dimension reduction

Since the time complexity of computing the skyline community increases with the number of dimensions, our intuition is to select a small number of important dimensions that can help to find the most important skyline communities. We here make use of the maximum entropy method to achieve this goal.

The detailed algorithm is given in Algorithm 4. In Algorithm 4, we first compute p dimensions that are crucial to community search, for some given parameter p. Specifically, the entropy for each combination of p dimensions are computed, and the one with the maximum entropy is selected (Lines 2 and 3). This p dimensions are denoted as P. Then Algorithm 3 is

Algorithm 3 MICS (*G, d, k,* dlist)

Input: A multi-valued graph $G = (V, E, A)$, the number of dimensions d, and a non-negative integer k
Output: Skyline communities of G

1 $R \leftarrow \varnothing$;
2 **if** $d = 2$ **then**
3 **return** BICS$(G, k,$ dlist$)$;
4 $G_1 \leftarrow$ the graph consisting of maximal k-cores in G;
5 $F \leftarrow$ the values on the d-th dimension;
6 $f^* \leftarrow$ MaxCI(G_1, d, k);
7 **foreach** $f \in F$ and $f \leqslant f^*$ **do**
8 dlist $\leftarrow$ dlist $\cup \{f\}$;
9 **foreach** $v \in G_1$ **do**
10 **if** $a_v^d < f$ **then**
11 $G_1 \leftarrow G_1 \setminus \{v\}$;
12 **if** $G_1 \neq \varnothing$ **then**
13 $R \leftarrow$ MICS$(G_1, d-1, k)$;
14 dlist $\leftarrow$ dlist $\setminus \{f\}$;
15 **return** R ;

Algorithm 4 P-MICS (*G, d, k,* dlist)

Input: A multi-valued graph $G = (V, E, A)$, the number of dimension d, and a non-negative integer k
Output: Skyline communities of G

1 plist $\leftarrow \varnothing$; $R \leftarrow \varnothing$;
2 Compute the entropy of all combinations of p dimensions in A;
3 $P \leftarrow$ Find the p dimensions with the maximum entropy;
4 $G_P \leftarrow G(V, E, P)$;
5 $R \leftarrow$ MICS(G_P, p, k, P);
6 **return** R ;

invoked to compute the skyline communities in the graph G_P which only consider the dimensions in P. As shown in Theorem 1, the skyline communities computed using graph G_P are also skyline communities of G.

As illustrated in Section 5, for a large k, Algorithm 4 can get almost all skyline communities of G. This means that Algorithm 4 can find almost all large communities with high influence. On the other hand, the following theorem shows that our algorithm can exponentially reduce the time complexity. The proof of Theorem 4 is similar to that of Theorem 3.

Theorem 4 The time complexity of Algorithm 4 is $O(l^{p-2}(l(m+n)))$.

5 Experimental Result

In this section, we evaluate our community search algorithm on real datasets. In particular, we will compare the efficiency and the effectiveness of algorithms MICS and P-MICS, i.e., the running time of these two algorithms and the quality of skyline communities these two algorithms find. All programs were implemented in Java language and compiled with IntelliJ IDEA, and all experiments were performed on a machine with Intel Core i5-7500 3.41 GHz and 8 GB DDR3-RAM in Windows 10.

We use the four real-world social network datasets listed in Table 1 to evaluate the performance of our algorithms. In Table 1, $d_{\max}$ and $k_{\max}$ represent the maximum degree and the maximum core number in the network, respectively. The datasets are downloaded from an interactive scientific network data repository (http://networkrepository.com). Besides, the generation of attribute values in the experiment is mainly subject to two distributions, uniform distribution, and normal distribution. In the experiments, the number of dimensions is set as $d = 10$.

5.1 Efficiency evaluation

The running time comparison of MICS and P-MICS is shown in Fig. 2. In Fig. 2, the running time corresponds to the case $d = 10$ for MICS, and the other two curves illustrate the running time for P-MICS in the cases that reducing the number of dimensions to $p = 6$ and $p = 3$. From Fig. 2, we can see that in all datasets, P-MICS can greatly reduce the running time. As shown in Fig. 3, which illustrates the reduction rate of P-MICS over MICS on the running time, it can be seen that when reducing the number of dimensions to $p = 3$, the running time can be reduced by around 30%, and when reducing the number of dimensions to $p = 3$, the reduction can be as large as 75%. Furthermore, it can also be found in Fig. 2 that the running time decreases promptly as k increases. This is because k limits the minimum degree of nodes in the target communities, and hence the larger k is, the more nodes are filtered out. In the dataset gowalla, the running time is less sensitive to k, due to its higher average degree.

Table 1 Network statistics.

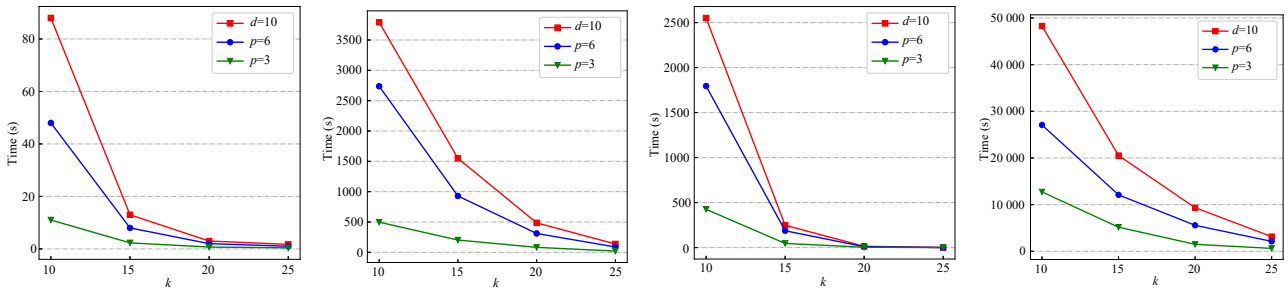
Dataset	Node	Edge	$d_{\max}$	$k_{\max}$
brightkite	5.7×10^4	2.13×10^5	10^3	53
gowalla	1.97×10^5	9.50×10^5	1.5×10^4	52
delicious	5.36×10^5	10^6	3×10^3	34
lastfm	1.2×10^6	4.5×10^6	5×10^3	71

The above experimental results show that our algorithms can efficiently compute the skyline communities, and P-MICS can greatly reduce the computation time. We next compare the quality of the skyline communities.

5.2 Quantity evaluation

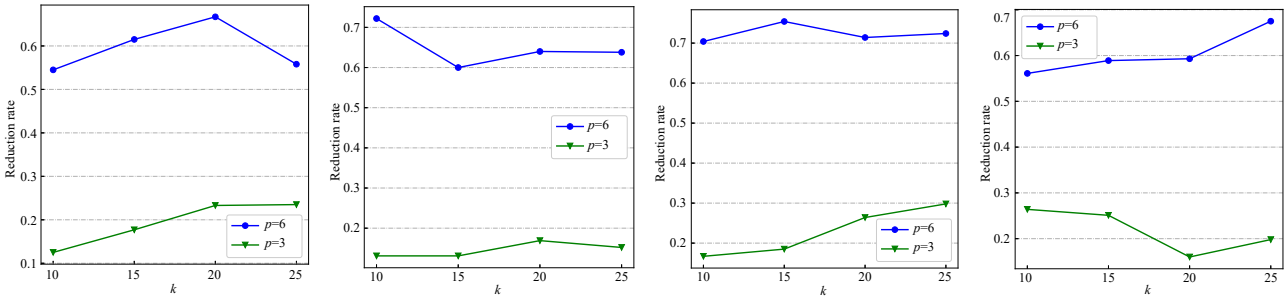
In Fig. 4, the numbers of skyline communities found by MICS ($d = 10$) and P-MICS ($p = 6$ and $p = 3$) are illustrated. From Fig. 4, it can be found that the more dimensions are considered in the algorithm, the more skyline communities can be found. However, when k is large, even if the number of dimensions is reduced to $p = 3$, P-MICS can find almost all skyline communities. This is because the maximum entropy method can select the most important dimensions that reserve essential information. On the other hand, the parameter k determines the cohesiveness of the skyline community. Hence, the communities with a large k are more important and are those that deserved to being detected.

To summarize, the experimental results show that our algorithms can efficiently and effectively find skyline communities. Comparing to MICS, by reducing the dimension but keeping the most important ones using the maximum entropy method, P-MICS can find almost all cohesive communities (i.e., those with a large k), while significantly reducing the computation time. Furthermore, the experimental results also imply that it is possible to find an appropriate dimension reduction parameter p to balance the reduction of running time and the quality of the community search.



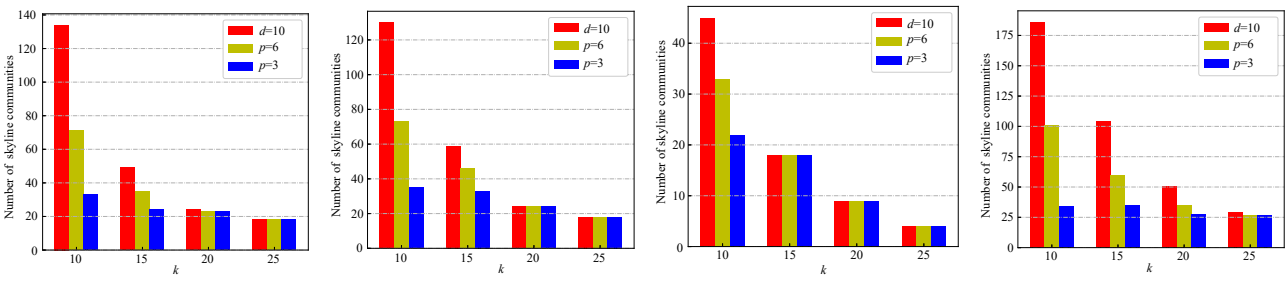
(a) brightkite (b) gowalla (c) delicious (d) lastfm

Fig. 2 Running time of MICS (d=10) and P-MICS (p=6 and p=3).



(a) brightkite (b) gowalla (c) delicious (d) lastfm

Fig. 3 Reduction rate of P-MICS comparing with MICS.



(a) brightkite (b) gowalla (c) delicious (d) lastfm

Fig. 4 Number of skyline communities found by MICS (d=10) and P-MICS (p=6 and p=3).

6 Conclusion

In this paper, we study how to find influential communities on a multi-valued graph. In our model, each node is associated with d attribute value, and the size of these values indicates its importance. What we are looking for are communities whose nodes are important and structure is cohesive. We propose the basic community search algorithm in the case of a high dimension, and we notice that the dimensions have a great influence on the algorithm time. Therefore, we then propose a dimensionality reduction algorithm with a maximum entropy model, which can preserve a lot of information even if fewer dimensions are considered. Experiments on real graphs also prove the effectiveness of our dimensionality reduction algorithm.

Acknowledgment

This work was partially supported by the National Key R&D Program of China (No. 2019YFB2102600) and the National Natural Science Foundation of China (Nos. 61971269, 61832012, 616727321, and 61771289).

References

[1] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[2] M. Sozio and A. Gionis, The community-search problem and how to plan a successful cocktail party, in *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2010, pp. 939–948.

[3] F. Zhang, Y. Zhang, L. Qin, W. J. Zhang, and X. M. Lin, When engagement meets similarity: Efficient (k, r)-core

computation on social networks, *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 998–1009, 2017.

[4] V. Batagelj and M. Zaversnik, An $O(m)$ algorithm for cores decomposition of networks, arXiv preprint arXiv: cs/0310049, 2003.

[5] S. Borzsony, D. Kossmann, and K. Stocker, The skyline operator, in *Proc. 17th Int. Conf. Data Engineering*, Heidelberg, Germany, 2001, pp. 421–430.

[6] K. Q. Zhang, H. Gao, X. X. Han, Z. P. Cai, and J. Z. Li, Modeling and computing probabilistic skyline on incomplete data, *IEEE Trans. Knowl. Data Eng*., doi: 10.1109/TKDE.2019.2904967.

[7] R. H. Li, L. Qin, F. H. Ye, J. X. Yu, X. K. Xiao, N. Xiao, and Z. B. Zheng, Skyline community search in multi-valued networks, in *Proc. 2018 Int. Conf. Management of Data*, Houston, TX, USA, 2018, pp. 457–472.

[8] Y. X. Fang, X. Huang, L. Qin, Y. Zhang, W. J. Zhang, R. Cheng, and X. M. Lin, A survey of community search over big graphs, arXiv preprint arXiv: 1904.12539, 2019.

[9] S. Fortunato, Community detection in graphs, arXiv preprint arXiv: 0906.0612, 2009.

[10] S. Fortunato and D. Hric, Community detection in networks: A user guide, arXiv preprint arXiv: 1608.00163, 2016.

[11] W. Y. Cui, Y. H. Xiao, H. X. Wang, and W. Wang, Local search of communities in large graphs, in *Proc. 2014 ACM SIGMOD Int. Conf. Management of Data*, Snowbird, UT, USA, 2014, pp. 991–1002.

[12] N. Barbieri, F. Bonchi, E. Galimberti, and F. Gullo, Efficient and effective community search, *Data Min. Knowl. Discov.*, vol. 29, no. 5, pp. 1406–1433, 2015.

[13] Y. X. Fang, Z. R. Wang, R. Cheng, H. Z. Wang, and J. F. Hu, Effective and efficient community search over large directed graphs (extended abstract), in *IEEE 35th Int. Conf. Data Engineering*, Macao, China, 2019, pp. 2157–2158.

[14] N. Wang, D. X. Yu, H. Jin, C. Qian, X. Xie, and Q. S. Hua, Parallel algorithm for core maintenance in dynamic graphs, in *Proc. 37th IEEE Int. Conf. Distributed Computing Systems*, Atlanta, GA, USA, 2017, pp. 2366–2371.

[15] H. Jin, N. Wang, D. X. Yu, Q. S. Hua, X. H. Shi, and X. Xie, Core maintenance in dynamic graphs: A parallel approach based on matching, *IEEE Trans. Parallel Distrib. Syst*., vol. 29, no. 11, pp. 2416–2428, 2018.

[16] Q. Luo, D. X. Yu, F. Li, Z. H. Dou, Z. P. Cai, J. G. Yu, and X. Z. Cheng, Distributed core decomposition in probabilistic graphs, in *Proc. 8th Int. Conf. Computational Data and Social Networks*, Ho Chi Minh City, Vietnam, 2019, pp. 16–32.

[17] Q. S. Hua, Y. L. Shi, D. X. Yu, H. Jin, J. G. Yu, Z. P. Cai, X. Z. Cheng, and H. H. Chen, Faster parallel core maintenance algorithms in dynamic graphs, *IEEE Trans. Parallel Distrib. Syst.,* vol. 31, no. 6, pp. 1287–1300, 2020.

[18] E. Akbas and P. X. Zhao, Truss-based community search: A truss-equivalence based indexing approach, *Proceedings of the PVLDB Endowment*, vol. 10, no. 11, pp. 1298–1309, 2017.

[19] X. Huang, H. Cheng, L. Qin, W. T. Tian, and J. X. Yu, Querying k-truss community in large and dynamic graphs, in *Proc. 2014 ACM SIGMOD Int. Conf. Management of Data*, Snowbird, UT, USA, 2014, pp. 1311–1322.

[20] Y. Wang, X. Jian, Z. H. Yang, and J. Li, Query optimal K-Plex based community in graphs, *Data Science and Engineering*, vol. 2, no. 4, pp. 257–273, 2017.

[21] D. N. Yang, Y. L. Chen, W. C. Lee, and M. S. Chen, On social-temporal group query with acquaintance constraint, *Proceedings of the PVLDB Endowment*, vol. 4, no. 6, pp. 397–408, 2011.

[22] L. Yuan, L. Qin, W. J. Zhang, L. J. Chang, and J. Y. Yang, Index-based densest clique percolation community search in networks (extended abstract), in *IEEE 35th Int. Conf. Data Engineering*, Macao, China, 2019, pp. 2161–2162.

[23] Y. X. Fang, R. Cheng, S. Q. Luo, and J. F. Hu, Effective community search for large attributed graphs, *Proceedings of the PVLDB Endowment*, vol. 9, no. 12, pp. 1233–1244, 2016.

[24] Y. X. Fang, R. Cheng, X. D. Li, S. Q. Luo, and J. F. Hu, Effective community search over large spatial graphs, *Proceedings of the PVLDB Endowment*, vol. 10, no. 6, pp. 709–720, 2017.

[25] K. Wang, X. Cao, X. M. Lin, W. J. Zhang, and L. Qin, Efficient computing of radius-bounded k-cores, in *IEEE 34th Int. Conf. Data Engineering*, Paris, France, 2018, pp. 233–244.

[26] Q. J. Zhu, H. B. Hu, C. Xu, J. L. Xu, and W. C. Lee, Geo-social group queries with minimum acquaintance constraints, *The VLDB Journal*, vol. 26, no. 5, pp. 709–727, 2017.

[27] R. H. Li, L. Qin, J. X. Yu, and R. Mao, Influential community search in large networks, *Proceedings of the PVLDB Endowment*, vol. 8, no. 5, pp. 509–520, 2015.

Lifang Zhang received the BS degree from Shandong University in 2019. She is currently pursuing for the master degree at Shandong University. Her research interests include graph analysis and data mining.

Qi Luo received the BS and MS degrees from Northeastern University in 2015, and Shandong University in 2018, respectively. He is currently pursuing the PhD degree at Shandong University. His research interests include graph analytics and distributed computing.

Dongxiao Yu received the BS degree from Shandong University in 2006, and the PhD degree from the University of Hong Kong in 2014. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology in 2016. Currently he is a professor in the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.

Xiuzhen Cheng received the MS and PhD degrees from the University of Minnesota–Twin Cities in 2000 and 2002, respectively. She is a professor in the School of Computer Science and Technology, Shandong University. Her current research interests include cyber physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time) and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is a fellow of IEEE and a member of ACM.

Jiguo Yu received the PhD degree from Shandong University in 2004. He became a full professor in the School of Computer Science, Qufu Normal University, Shandong, China in 2007. Currently he is a full professor in Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), and a professor in the School of Information Science and Engineering, Qufu Normal University. His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. Particularly, he is interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a senior member of IEEE, a member of ACM, and a senior member of the China Computer Federation (CCF).

Zhipeng Cai received the MS and PhD degrees from University of Alberta in 2004 and 2008, respectively. He is currently an associate professor in the Department of Computer Science, Georgia State University (GSU). Prior to joining GSU, he was a research faculty in the School of Electrical and Computer Engineering, Georgia Institute of Technology. His research areas focus on networking, big data, data security, and artificial intelligence. He is the recipient of an NSF CAREER Award.