Top-k Community Similarity Search Over Large Road-Network Graphs

Niranjan Rai
Department of Computer Science
Kent State University
Kent, OH 44242, USA
nrai@kent.edu

Xiang Lian
Department of Computer Science
Kent State University
Kent, OH 44242, USA
xlian@kent.edu

***Abstract*—With the urbanization and development of infrastructure, the community search over road networks has become increasingly important in many real applications such as urban/city planning, social study on local communities, and community recommendations by real estate agencies. In this paper, we propose a novel problem, namely *top-k community similarity search* ($Top\text{-}kCS^2$), which efficiently and effectively obtains spatial communities that are the most similar to a given query community over road-network graphs. In order to efficiently and effectively tackle the $Top\text{-}kCS^2$ problem, in this paper, we will design an effective similarity measure between communities, and propose a framework for retrieving $Top\text{-}kCS^2$ query answers. Extensive experiments have been conducted on real and synthetic data sets to confirm the efficiency and effectiveness of our proposed $Top\text{-}kCS^2$ approach under various parameter settings.**

***Index Terms*—top-k community similarity search, road-network graph**

I. Introduction

Recently, the community search/detection over graphs has received much attention in many real-world applications such as social network analysis [1]–[11], online marketing and advertising over geo-social networks [12]–[17], and many others. While prior works on the community search/detection [7]–[9], [11], [18]–[21] usually considered *user communities* with strong social/spatial relationships in (geo-)social networks, in this paper, we will study a novel problem of retrieving top-k spatial communities on road-network graphs, which are quite useful and important for urban/city planning or community recommendations by real estate agencies.

We have the following motivation example.

Example 1. (Data Visualization via Lenses on Road Networks) *In real applications such as urban/city planning, social study, or transportation systems, data analysts often utilize geospatial visualization tools such as interactive lens [22] and identify/analyze those communities with neighborhood similar to a target (query) community. Figure 2 illustrates a map of road networks in a visualization system, on which a lens (i.e., a circle with radius r) is specified by a user. The road-network subgraph within the lens can be considered as a query community, which is used to find other communities nearby with similar road-network structures and POIs. Geologists or other data analysts may be interested in studying community structure in a particular spatial location (e.g. a city or a country), our solution will help find/highlight different community structures in a specific location based on a given query community.* ■

Fig. 1. An example of lens on a road-network graph G.

Inspired by the examples above, in this paper, we will formulate and tackle a novel problem, namely *top-k community similarity search* ($Top\text{-}kCS^2$), which efficiently and effectively obtains top-k spatial communities that are similar and spatially close to a given query community over road-network graphs.

Note that, efficient and effective answering of the $Top\text{-}kCS^2$ query is rather challenging. A straightforward method to process the $Top\text{-}kCS^2$ query is to enumerate all possible communities (subgraphs) in road-network graphs, compute the similarity/distance between each community and the query community, and return k communities with higher similarities than a given threshold, θ and small distances. However, this straightforward method is not very efficient, due to the large number of candidate communities to refine on road networks. What is more, it is not very trivial how to accurately define the similarity between two communities that captures their graph structural and POI similarities.

To our best knowledge, prior works (e.g., community search in (geo-)social networks) did not consider finding

978-1-7281-9184-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDE51399.2021.00206

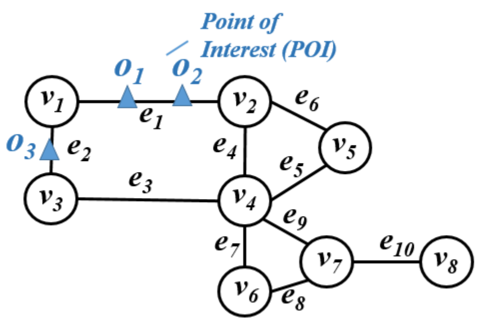


Fig. 2. An example of a road-network graph G.

similar/close spatial communities on large-scale road-network graphs. Therefore, previous techniques cannot be directly applied to solve our $Top\text{-}kCS^2$ problem. In order to tackle the challenges of processing $Top\text{-}kCS^2$ queries, in this paper, we will propose a novel metric to measure the similarity between two spatial communities in road-network graphs, design effective pruning strategies (w.r.t. similarity and distance) to reduce the $Top\text{-}kCS^2$ search space, as well as an effective indexing mechanism, and develop an efficient $Top\text{-}kCS^2$ processing algorithm via index that integrates our proposed pruning methods.

In this paper, we make the following contributions.

- We formally define a novel problem, namely *top-k community similarity search* ($Top\text{-}kCS^2$) query, over road-network graphs in Section II.
- We present the framework for answering the $Top\text{-}kCS^2$ query in Section III.
- We demonstrate the efficiency and effectiveness of our proposed $Top\text{-}kCS^2$ approach in Section IV.

In addition, Section V concludes this paper.

II. Problem Definition

In this section, we formally define *top-k community similarity search over road-network graphs*.

A. Road-Network Graphs

Road Networks. In this paper, we model road networks by a planar graph, defined as follows.

Definition 1. ***(Road-Network Graph)*** *A road-network graph is a connected planar graph $G = (V(G), E(G), \Phi(G))$, where $V(G)$ and $E(G)$ are the sets of vertices and edges in graph G, respectively, and $\Phi(G)$ is a mapping function: $V(G) \times V(G) \rightarrow E(G)$.*

In Definition 1, edges $e_i \in E(G)$ represent roads (line segments) in road networks G, where each edge e_i is associated with its length $e_i.l$. Moreover, vertices $v_i \in V(G)$ correspond to intersection points of road segments.

Example 2. *Figure 2 shows an example of a road-network graph G, where the vertex set $V(G) = \{v_1, v_2, ..., v_8\}$ and the edge set $E(G) = \{e_1, e_2, ..., e_{10}\}$. For example, edge e_1 is a road segment connecting 2 ending vertices v_1 and v_2.* ■

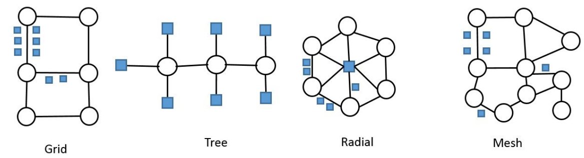


Fig. 3. An example of patterns in road-network graphs.

Points of Interest. On edges $e_i \in E(G)$ of road networks G, there are a number of *points of interest* (POIs), such as restaurants and movie theatres, which are defined as follows:

Definition 2. ***(Points of Interest, POI)*** *Given a road-network graph G, a point of interest (POI), o_j, is a facility (object) located at $o_j.loc$ on an edge $e_i \in E(G)$.*

In Definition 2, POIs on edge $e \in E(G)$ can be of various types, such as restaurants, shopping malls, supermarkets, cinemas, schools, churches, houses, and so on. We can represent all POIs on edge $e \in E(G)$ by a POI vector, $e.vec$, which consists of counts (frequencies) of different POI types on edge e. For example, assume that we only consider 4 types of POIs, restaurant, church, house, and school. If an edge e contains 2 restaurants, 1 church, 3 houses, and 5 schools, then its POI vector $e.vec$ is given by $e.vec = (2, 1, 3, 5)$.

Example 3. *As illustrated in Figure 2, there are two POI objects o_1 and o_2 on edge e_1, where object o_1 represents a house and object o_2 is a school. Thus, the POI vector, $e_1.vec$, of edge e_1 is given by $(1, 1)$.* ■

B. The Spatial Community in Road-Network Graphs

Before we define the spatial community on road networks, we first discuss *patterns* and *unit patterns* in road-network graphs.

Road-Network Patterns. As shown in Figure 3, there are many possible structural patterns in road-network graphs, which may indicate different scenarios of road-network designs. For example, in city areas, it is very likely that we have a large number of *grid* structures of rectangular shape, due to the block system planning. Note that, the grid pattern has a lot of intersections and short roads, and a study [23] has shown that the number of accidents is higher for this pattern type than others.

The *tree* pattern is another common pattern found on road networks. This type of pattern is very common in residential areas, where houses, churches, and/or schools are usually located at the leaves of trees.

The *radial* pattern is composed of a network of roads, which radiate from a core. Such a pattern type usually indicates a business area.

The *mesh* pattern is usually the results of unplanned road networks, where there are a lot of structures of different pattern types.

Unit Patterns in Road-Network Graphs. Figure 4 illustrates several basic patterns, called *unit patterns*, on road networks, which include *edge*, *delta*, *rectangle*, *pentagon*, *hexagon*, and

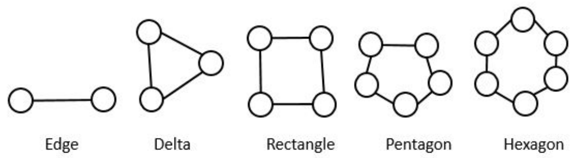


Fig. 4. An example of unit patterns in road-network graphs.

so on. In particular, the *edge* unit pattern is an edge (a road segment, but not in a circle), which can be a branch in the tree pattern or a dead end in residential areas. Similarly, the *delta* unit pattern contains 3 vertices, forming a circular triangular structure.

In this paper, we consider road networks as a planar graph. Thus, we can divide this planar road-network graph into non-overlapping unit patterns of different types. Intuitively, unit patterns such as rectangles correspond to blocks on road networks.

Spatial Community. Next, we give the definition of a spatial community in road-network graphs.

Definition 3. ***(Spatial Community in a Road-Network Graph)****. Given a road-network graph G, a center vertex $v_c \in V(G)$, and a radius r, a spatial community, C_l, is a subgraph of G (i.e., $C_l \subseteq G$), such that:*

1) *subgraph C_l is connected, and;*
2) *all unit patterns c in C_l have the minimum distances to vertex v_c less than or equal to r (i.e., $mindist(c, v_c) \leq r$),*

where $mindist(c, v_c)$ computes the minimum Euclidean distance from vertex v_c to unit pattern c.

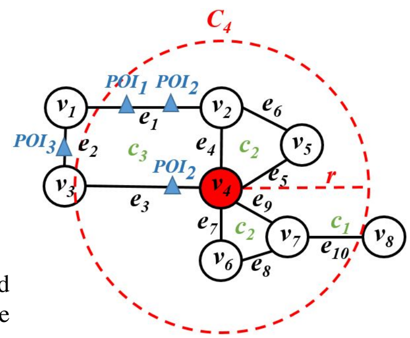
Intuitively, a spatial community is a subgraph of road networks G, whose unit patterns (i.e., blocks) intersect with a circle centered at vertex v_c and with radius r.

Note that, in this paper, we assume that radius r is a pre-defined system parameter, which can be specified by the system (e.g., the radius of lens in a visualization system) or tuned/inferred from historical users' preferences (e.g., 10 miles within users' driving distances).

Example 4. *In the example of Figure 5(a), assume that we have a center vertex v_4, and a radius r. Then, a spatial community, C_4, centered at vertex v_4 and with radius r, is given by a subgraph with vertices $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ and edges $e_1 \sim e_{10}$. Note that, edge e_2 is considered to be inside the community C_4, since it is a part of the rectangle unit pattern (i.e., $\square v_1 v_2 v_3 v_4$), denoted as c_3, which partially intersects with a circle centered at vertex v_4 and with radius r.* ■

C. Similarity Between Two Communities

In this subsection, we first propose a similarity metric to measure the similarity between two unit patterns, and then provide the definition of the similarity score between two communities.



(a) A spatial community C_4

C_4	POI_1	POI_2	POI_3	POI_4
$c_1[1].vec$	2	1	1	0
$c_2[1].vec$	2	5	0	2
$c_2[2].vec$	4	3	1	2
$c_3[1].vec$	1	2	1	0

(b) POI vector of C_4

Fig. 5. An example of a spatial community.

The Similarity Score Between Unit Patterns. We first give the definition of the similarity score between two unit patterns. In particular, for two unit patterns of the same type (e.g., delta or rectangle), we define their similarity based on their POIs via the *cosine similarity* [24].

Definition 4. ***(The Similarity Score of Two Unit Patterns)*** *Assume that we have two unit patterns c_x and c_y, whose POI vectors are represented by $c_x.vec$ and $c_y.vec$, respectively. Then, we can compute their similarity score as:*

$$sim(c_x, c_y) = cos\_sim(c_x.vec, c_y.vec), \tag{1}$$

where function $cos\_sim(c_x.vec, c_y.vec)$ outputs the cosine similarity [24] between vectors $c_x.vec$ and $c_y.vec$.

In particular, given two vectors $A = (A_1, A_2, \cdots, A_n)$ and $B = (B_1, B_2, \cdots, B_n)$, the cosine similarity, $cos\_sim(A, B)$, in Eq. (1) is given by the normalized dot product of vectors A and B as follows:

$$cos\_sim(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{h=1}^{n} A_h B_h}{\sqrt{\sum_{h=1}^{n} A_h^2}\sqrt{\sum_{h=1}^{n} B_h^2}}. \tag{2}$$

Note that, in Eq. (2), we assume that vectors A and B (or POI vectors of unit patterns in Eq. (1)) are normalized to have length 1 (i.e., $||A|| = ||B|| = 1$). As a result, we have:

$$\begin{aligned} cos\_sim(c_x.vec, c_y.vec) &= c_x.vec \cdot c_y.vec \qquad (3) \\ &= \sum_{h=1}^{n} \big(c_x.vec[h] \times c_y.vec[h]\big). \end{aligned}$$

For example, assume that we have a query unit pattern (edge) $q_1[1]$, whose POI vector is given by $q_1[1].vec$ = $(2, 2, 1, 1)$. From Figure 5, we have $c_1[1]$ which is an edge similar to $q_1[1]$, where $c_1[1].vec = (2, 1, 1, 0)$. Now, the similarity score between unit pattern $c_1[1].vec$ and $q_1[1].vec$ can be calculated as $cos\_sim(q_1[1].vec, c_1[1].vec)$, which is equal to 7 ($= 2 \times 2 + 2 \times 1 + 1 \times 1 + 1 \times 0$).

The Similarity Score Between Spatial Communities. The similarity score between a candidate community C_l and a query community, Q, can be calculated below.

Definition 5. ***(The Similarity Score Between Two Communities).*** *Given a community C_l, a query community, Q, and their unit patterns (of the h-th type) $c_h \in C_l$ and $q_h \in Q$ $(1 \le h \le n)$, the similarity score, $sim(C_l, Q)$, between communities C_l and Q is given by the average cosine similarity of POI vectors of each unit pattern type in c_h and q_h, that is,*

$$\begin{aligned} & sim(C_l, Q) \qquad (4) \\ = & \frac{\sum_{h=1}^{n} sim(c_h, q_h)}{n} \\ = & \sum_{h=1}^{n} \frac{\left\{ \frac{\sum_{i=1}^{|c_h|} \sum_{j=1}^{|q_h|} cos\_sim(c_h[i].vec, q_h[j].vec)}{|q_h|} \right\}}{n} \\ = & \sum_{h=1}^{n} \frac{\sum_{i=1}^{|c_h|} \sum_{j=1}^{|q_h|} cos\_sim(c_h[i].vec, q_h[j].vec)}{|q_h| \cdot n}, \end{aligned}$$

where $|q_h|$ is the number of unit patterns of the h-th shape in the query community Q, and $c_h[i]$ (or $q_h[j]$) is the i-th (or j-th) unit pattern (with the h-th shape) in C_l and Q, respectively.

Intuitively, the community C_l may contain n unit pattern types, the h-th of which may have $|c_h|$ instances of such a unit pattern type in C_l. The case of the query community Q is similar. Thus, in Eq. (4), for the h-th unit pattern type, we can compute the summed similarity between unit patterns $c_h[i]$ and $q_h[j]$, divide it by $|q_h|$, and then take the average score for all the n unit pattern types.

As an example, in Figures 5 and 6, we have a candidate community C_4 and a query community Q, respectively. Here, C_4 and Q have three types of unit patterns, edge (c_1), delta (c_2) and rectangle (c_3), with counts $(1, 2, 1)$ and $(1, 1, 2)$, respectively. Thus, based on Eq. (4), the similarity score $sim(C_4, Q)$ can be calculated as the average similarity of all three unit pattern types in C_4 and Q, that is, $\frac{sim(c_1,q_1)+sim(c_2,q_2)+sim(c_3,q_3)}{3}$.

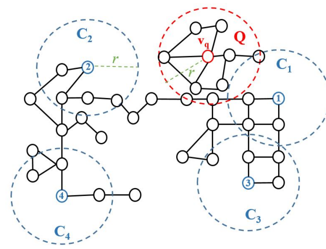
D. Top-k Community Similarity Search in Road-Network Graphs

We next define the problem of *top-k community similarity search* ($Top\text{-}kCS^2$).

Definition 6. ***(Top-k Community Similarity Search in Road-Network Graphs,*** $Top\text{-}kCS^2$***)*** *Given a query community Q, a road-network graph G, a query vertex v_q, and a similarity threshold θ, a top-k community similarity search ($Top\text{-}kCS^2$) query retrieves k communities, C_l (for $1 \le l \le k$), from G, such that:*

- *similarity scores $sim(C_l, Q)$ are greater than or equal to θ (i.e., $sim(C_l, Q) \ge \theta$, and;*
- *for any community C_j (satisfying $sim(C_j, Q) \ge \theta$ and $C_j \ne C_l$), we have $dist(v_q, C_l) < dist(v_q, C_j)$ (i.e., communities C_l are the closest to v_q)*

where the distance from v_q to a community C_l is given by the Euclidean Distance [25] between center vertices, v_q and v_c, from communities Q and C_l, respectively.



Q	POI_1	POI_2	POI_3	POI_4
$q_1[1].vec$	1	2	1	1
$q_2[1].vec$	3	3	3	3
$q_3[1].vec$	2	3	0	2
$q_3[2].vec$	2	2	1	1

Fig. 6. An example of communities in a large road-network graph.

As an example in Figure 6, we have a query community Q, a query vertex, v_q, and a radius r. In the figure, we have some candidate communities $\{C_1, C_2, C_3, C_4\}$. Assume that the similarity scores of communities C_1, C_2, C_3, and C_4 are 0.7, 0.5, 0.35 and 0.5, respectively. Moreover, the distances (in miles) from v_q to communities C_1, C_2, C_3, and C_4 are 0.6, 0.2, 0.55, and 0.4, respectively. If the similarity threshold θ is 0.5 and $k = 1$, then the $Top\text{-}kCS^2$ problem will return C_2 as the answer. This is because the similarity score between C_2 and Q is greater than or equal to 0.5 (i.e., θ) and community C_2 has the smallest distance to v_q among communities $C_1 \sim C_4$.

Table I depicts the commonly-used notations in this paper and their descriptions.

TABLE I
NOTATIONS AND DESCRIPTIONS

Notation	Description
o	a point of interest (POI)
c	a unit pattern
C_l	a spatial community in road-network graph G
Q	a given query community
n	the total no. of unit pattern types
c_h	a unit pattern of type h $(1 \le h \le n)$ in community C_l
q_h	a unit pattern of type h in query community, Q
$c_h[i].vec$	a POI vector for i-th unit pattern of c_h $(1 \le i \le \vert c_h \vert)$
$q_h[j].vec$	a POI vector for j-th unit pattern of q_h $(1 \le i \le \vert q_h \vert)$
$\vert c_h \vert$	the count of the unit pattern of type h in C_l
$\vert q_h \vert$	the count of the unit pattern of type h in Q

III. The Framework for $Top\text{-}kCS^2$ Query Answering

In this section, we present a framework for $Top\text{-}kCS^2$ query answering in road-network graphs, G, in Algorithm 1.

Specifically, our framework helps retrieve top-k similar communities that satisfy the similarity threshold when compared to the query community, Q, and are closer to the query vertex, v_q, which consists of *offline pre-processing* and *online computation phases*.

In the *offline pre-processing phase*, we first detect all the unit patterns, c, on road networks G (line 1), by invoking our proposed algorithm, $Get\_Unit(G)$. Then, we insert all the unit patterns, $c \in G$, into an aggregate R-tree index, I, that is, aR-tree [26], and offline pre-compute all the communities (with radius r) in the road-network graph G, whose statistics

Algorithm 1: The $Top\text{-}kCS^2$ Answering Framework

```
   Input: a road-network graph G, a similarity threshold θ, radius r, a query
          community Q, and a query vertex v_q
   Output: top-k communities, C_l, (1 ≤ l ≤ k) with similarity scores ≥ θ
   //Offline Pre-Processing Phase
1  detect all the unit patterns c ∈ G
2  insert all the unit patterns into an aR-tree I
3  obtain some communities C_l (1 ≤ l ≤ |V|) containing each unit pattern and
     update aggregates in aR-tree I
   //Online Computation Phase
4  for each unit pattern type q_h ∈ Q, 1 ≤ h ≤ n do
5      for each unit pattern q_h[i], 1 ≤ i ≤ |q_h| do
6          find a set of unit patterns similar to q_h[i] via index I

7  sort candidate unit patterns based on their similarity scores
8  obtain a list of candidate communities, cand_list, based on the sets of
     candidate unit patterns w.r.t. q_h ∈ Q
9  for each candidate community C_l ∈ cand_list do
10     calculate an upper bound, ub_sim(C_l, Q), of the similarity score
         sim(C_l, Q)
11     if ub_sim(C_l, Q) < θ then
12         prune community C_l
13     else
14         calculate the exact score of C_l, sim(C_l, Q)
15         if sim(C_l, Q) < θ then
16             prune community C_l
17         else
18             if comm_count < k then
19                 add C_l to a sorted top-k list ans_list
20                 comm_count++
21             else
22                 if dist(C_l, v_q) < dist(C_k, v_q) then
23                     add C_l to the top-k list ans_list
24                     remove C_k from the top-k list ans_list

25 return the top-k answer list ans_list
```

(e.g., lower/upper bounds of pattern counts) can be used as aggregates for unit patterns in the aR-tree (lines 2-3).

In the *online computation phase*, for each unit pattern q_h in the query community Q, we use the aR-Tree to retrieve a set of similar candidate unit patterns, in descending order of similarity scores (lines 4-7). Next, we use these candidate unit patterns, with respect to q_h $(1 \leq h \leq n)$, to obtain a number of candidate communities C_l in a list $cand\_list$ (line 8). For each candidate community, $C_l \in cand\_list$, we first calculate the upper bound similarity score, $ub\_sim(C_l, Q)$ (lines 9-10). If the similarity upper bound score of C_l is less than threshold θ (i.e. $ub\_sim(C_l, Q) < \theta$), we can safely prune the community C_l (lines 11-12). Otherwise, we calculate the exact similarity score, $sim(C_l, Q)$, for candidate community C_l (line 14). If it holds that $sim(C_l, Q) < \theta$, then we can safely rule out community C_l (lines 15-16). On the other hand, if $sim(C_l, Q) \geq \theta$ holds, we will check whether we have k candidate communities in the current top-k list, ans_list (lines 17-24). When the count, $comm\_count$, of communities in the current top-k list ans_list is less than k, we can directly add community C_l to this list and increase the count, $comm\_count$, by 1 (lines 18-20). When $comm\_count$ is equal to k, we will consider the constraint of the distance of community C_l to query vertex v_q. If C_l is closer than the k-th closest community C_k in the top-k list ans_list, then we remove community C_k from the list and insert C_l into the top-k list ans_list (lines 22-24). Finally, we return the top-k answer list, ans_list, after checking all candidate communities in $cand\_list$ (line 25).

IV. Experimental Evaluation

In this section, we verify the effectiveness and efficiency of our proposed $Top\text{-}kCS^2$ algorithm over both real and synthetic road-network graphs.

A. Experimental Settings

Real/synthetic data sets. We used both real and synthetic data sets for our experimental evaluation. Specifically, for real data set, we use the California Road Network [27], denoted as CA, which contains 21,048 road intersection points, 21,693 road segments, and 104,770 *points of interests* (POIs). CA is originally obtained from Digital Chart of the World Server and U.S. Geological Survey. Each vertex in CA data set is represented by (longitude, latitude).

For synthetic data, we first generate vertices of a road-network graph on a spatial data space, following either the Uniform or Clustered distribution. For the Uniform distribution, we generate vertices uniformly in a designated spatial data space; for the clustered data set, we first randomly obtain seed vertices in a spatial space, and then generate other vertices close to these seeds. Here, the clustered data set can simulate dense road networks (i.e., clusters of vertices) in cities. Next, we connect vertices via edges (road segments) on road networks, that is, linking each vertex to $d \in [deg_{min}, deg_{max}]$ random nearest neighbors nearby (avoiding road intersections on the planar graph). This way, we can obtain a random road-network graph, G, with an average degree deg. By using different spatial distributions of vertices, we produce two types of graph, $uniform$ and $cluster$.

Measures. To evaluate the $Top\text{-}kCS^2$ query performance, we select 15 random query vertices from road networks, and obtain query communities (with radius r). We report the *wall clock time* and *I/O cost*. Here, the *wall clock time* is the average time cost to answer $Top\text{-}kCS^2$ queries; the *I/O cost* is the number of node accesses in the aR-tree.

Competitor. To our best knowledge, no prior works studied the top-k community search problem in large-scale road-network graphs, which has different community semantics from that on social networks. Thus, in this paper, we compare our $Top\text{-}kCS^2$ approach with a baseline algorithm, named $baseline$, which is a naive approach without using any index. In particular, the $baseline$ method first scans the road-network graph G to retrieve unit patterns from G that are similar to query unit patterns in the query community Q, and then computes k communities (containing the retrieved unit patterns) that satisfy the similarity threshold, θ and are closest to Q.

Parameter settings. Table II depicts the parameter settings, where default values are in bold. Each time we vary the values of one parameter, while other parameters are set to their default values. We ran all the experiments on a machine with Intel Core i7-6600U 2.60GHz CPU, Windows 10 OS, and 512 GB memory. All algorithms were implemented in C++.

The $Top\text{-}kCS^2$ performance vs. real/synthetic data sets. Figure 7 compares our $Top\text{-}kCS^2$ approach with the $baseline$

TABLE II
THE PARAMETER SETTINGS.

Parameters	Values
k	1, 5, **10**, 15, 20
deg	2, **3**, 4
r	0.1, 0.5, **1**, 1.5, 2
θ	0.5, 0.55, **0.6**, 0.65, 0.7
$\|V(G)\|$	10K, 20K, **30K**, 50K, 100K

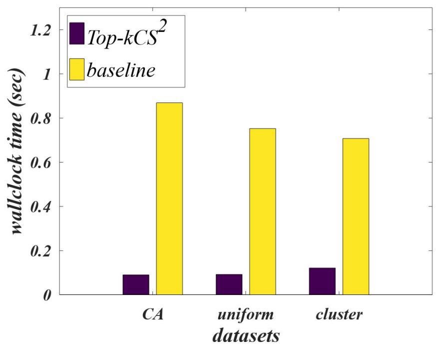


Fig. 7. The $Top\text{-}kCS^2$ performance vs. real/synthetic data sets.

algorithm over real/synthetic data sets, in terms of the wall clock time. From the figure, we can see that the efficiency of the $Top\text{-}kCS^2$ query outperforms that of $baseline$ for all the three data sets. This is because $Top\text{-}kCS^2$ applies effective pruning methods with the help of the index. The experimental results confirm the effectiveness of our pruning methods, and efficiency of our $Top\text{-}kCS^2$ approach.

V. CONCLUSIONS

In this paper, we formulate and tackle a novel problem of *top-k community similarity search* ($Top\text{-}kCS^2$) over large-scale road-network graphs, which retrieves k spatial communities having high structural and POI similarities and with spatial closeness, with respect to a given query community. To tackle this problem, we propose effective pruning strategies and indexing mechanism, and develop an efficient $Top\text{-}kCS^2$ query processing algorithm. We have demonstrated through extensive experiments the efficiency and effectiveness of our proposed $Top\text{-}kCS^2$ approach over both real and synthetic road-network graphs.

ACKNOWLEDGEMENTS

Xiang Lian was supported by NSF OAC (No. 1739491) and Lian Startup (No. 220981) from Kent State University.

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, 2010.

[2] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, 2004.

[3] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "A model-based approach to attributed graph clustering," in *Proceedings of the 2012 ACM SIGMOD*, 2012.

[4] Y. Liu, A. Niculescu-Mizil, and W. Gryc, "Topic-link lda: joint models of topic and author community," in *ICML*, 2009.

[5] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen, "Joint latent topic models for text and citations," in *Proceedings of the 14th ACM SIGKDD*, 2008.

[6] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *PVLDB*, vol. 2, no. 1, 2009.

[7] M. Sozio and A. Gionis, "The community-search problem and how to plan a successful cocktail party," in *Proceedings of the 16th ACM SIGKDD*, 2010.

[8] W. Cui, Y. Xiao, H. Wang, and W. Wang, "Local search of communities in large graphs," in *Proceedings of the 2014 ACM SIGMOD*, 2014.

[9] W. Cui, Y. Xiao, H. Wang, Y. Lu, and W. Wang, "Online search of overlapping communities," in *Proceedings of the 2013 ACM SIGMOD*, 2013.

[10] R.-H. Li, L. Qin, J. X. Yu, and R. Mao, "Influential community search in large networks," *PVLDB*, vol. 8, no. 5, 2015.

[11] X. Huang, L. V. Lakshmanan, J. X. Yu, and H. Cheng, "Approximate closest community search in networks," *arXiv preprint arXiv:1505.05956*, 2015.

[12] D.-N. Yang, C.-Y. Shen, W.-C. Lee, and M.-S. Chen, "On socio-spatial group query for location-based social networks," in *Proceedings of the 18th ACM SIGKDD*, 2012.

[13] Y. Li, D. Wu, J. Xu, B. Choi, and W. Su, "Spatial-aware interest group queries in location-based social networks," in *ICIK*, 2012.

[14] Y. Yuan, X. Lian, L. Chen, Y. Sun, and G. Wang, "Rs$k$nn: $k$nn search on road networks by incorporating social influence," *IEEE TKDE*, vol. 28, no. 6, 2016.

[15] Y. Fang, R. Cheng, X. Li, S. Luo, and J. Hu, "Effective community search over large spatial graphs," *PVLDB*, vol. 10, no. 6, 2017.

[16] A. Al-Baghdadi, G. Sharma, and X. Lian, "Efficient processing of group planning queries over spatial-social networks," *IEEE TKDE*, 2020.

[17] L. Chen, C. Liu, R. Zhou, J. Li, X. Yang, and B. Wang, "Maximum co-located community search in large scale social networks," *PVLDB*, vol. 11, no. 10, 2018.

[18] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, 2010.

[19] M. Newman et al., "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, 2004.

[20] R. Li, L. Qin, J. X. Yu, and R. Mao, "Influential community search in large networks," *Proc. VLDB Endow.*, vol. 8, no. 5, 2015.

[21] Y. Fang, R. Cheng, S. Luo, and J. Hu, "Effective community search for large attributed graphs," *Proc. VLDB Endow.*, vol. 9, no. 12, 2016.

[22] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann, "Interactive lenses for visualization: An extended survey," *Computer Graphics Forum*, vol. 36, 05 2016.

[23] *Roundabouts and Mini Roundabouts*, U.S. Department of Transportation, February 2020, https://safety.fhwa.dot.gov/intersection/innovative/roundabouts/.

[24] W. contributors, "Cosine similarity — Wikipedia, the free encyclopedia," 2018, [Online; accessed 18-September-2018].

[25] Wikipedia contributors, "Euclidean distance — Wikipedia, the free encyclopedia," 2020, [Online; accessed 5-February-2020].

[26] I. Lazaridis and S. Mehrotra, "Progressive approximate aggregate queries with a multi-resolution tree structure," *Acm sigmod record*, vol. 30, no. 2, 2001.

[27] "Real datasets for spatial databases: Road networks and points of interest." [Online]. Available: https://www.cs.utah.edu/lifeifei/SpatialDataset.htm