# Geo-Social Influence Spanning Maximization

Jianxin Li, Timos Sellis, *Fellow, IEEE*, J. Shane Culpepper, Zhenying He, Chengfei Liu, *Member, IEEE*, and Junhu Wang

**Abstract**—Influence maximization is a recent but well-studied problem which helps identify a small set of users that are most likely to "influence" the maximum number of users in a social network. The problem has attracted a lot of attention as it provides a way to improve marketing, branding, and product adoption. However, existing studies rarely consider the physical locations of the users, but location is an important factor in targeted marketing. In this paper, we propose and investigate the problem of influence maximization in location-aware social networks, or, more generally, *Geo-social Influence Spanning Maximization*. Given a query $q$ composed of a region $R$, a regional acceptance rate $\rho$, and an integer $k$ as a seed selection budget, our aim is to find the maximum geographic spanning regions (MGSR). We refer to this as the MGSR problem. Our approach differs from previous work as we focus more on identifying the maximum spanning geographical regions within a region $R$, rather than just the number of activated users in the given network like the traditional influence maximization problem [14]. Our research approach can be effectively used for online marketing campaigns that depend on the physical location of social users. To address the MGSR problem, we first prove NP-Hardness. Next, we present a greedy algorithm with a $1 - 1/e$ approximation ratio to solve the problem, and further improve the efficiency by developing an upper bounded pruning approach. Then, we propose the OIR*-Tree index, which is a hybrid index combining ordered influential node lists with an R*-tree. We show that our index based approach is significantly more efficient than the greedy algorithm and the upper bounded pruning algorithm, especially when $k$ is large. Finally, we evaluate the performance for all of the proposed approaches using three real datasets.

**Index Terms**—Spatial-social influence, influence spanning maximization, social media

## 1 Introduction

Influence maximization has attracted considerable attention in recent years as a means of enhancing marketing campaigns through social networks [7], [8], [11], [12], [14], [15]. In real applications, companies often run surveys for new products, and perform market tests via social networks before they produce the new products. In general, these companies are only able to choose a small number of social users to trial their new products due to financial restrictions. So, the question is: Which users should be targeted by the companies if the goal is to seed the social network such that as many users as possible will hear about the products? Or simply, what set of users can exert the most influence over the entire network? This well studied problem is commonly referred to as *Influence Maximization*.

However, there is a significant and open problem in current marketing efforts: How do you select a certain number of users whose influence can maximally cover an agent's targeted geographical region? For example, consider political campaign engagement via social networking. Social media has been a powerful engagement tool in recent political campaigns such as the US Election [22], the UK Election [1], and the Australian Election [5]. Under normal circumstances, a politician wants to target conversations in a particular region. To do this, the approach is to select or persuade a certain number of social users who have the maximum influence coverage within the targeted region. Inside the targeted region, many small sub-regions may exist which require more attention than others. The candidate wants to influence more sub-regions where a minimum percentage of votes is needed in each sub-region in order to win the election. Therefore, the goal is to influence the maximum number of sub-regions wherein each sub-region there is a certain percentage of social users who can be influenced. There are other social network driven campaigns in real world scenarios such as physical advertisement placement, new service facility placement, and sporting events, that might benefit from this line of research.

Consider another example where *Adidas-Australia* would like to launch a citywide marathon within the *greater Melbourne region*. To attract more participants and advertise their products, *Adidas-Australia* would like to carefully select and plan a starting venue, a stop venue, as well as running routes within the targeted *greater Melbourne region*. To address this issue, our proposed technique can recommend a certain number ($k$) of influential users and their influenced sub-regions to *Adidas-Australia*. The selected users have the maximum influence spanning inside the

- *J. Li is with the School of Computer Science and Software Engineering, University of Western Australia, Crawley, WA 6009, Australia. E-mail: jianxin.li@uwa.edu.au.*
- *T. Sellis and C. Liu are with the Faculty of Science, Engineering and Technology, Swinburne University of Technology, Hawthorn, Vic 3122, Australia. E-mail: {tsellis, cliu}@swin.edu.au.*
- *J.S. Culpepper is with the School of Computer Science and Information Technology, RMIT, Melbourne, Vic 3000, Australia. E-mail: shane.culpepper@rmit.edu.au.*
- *Z. He is with the School of Computer Science, Fudan University, Shanghai Shi 200433, China. E-mail: zhenying@fudan.edu.cn.*
- *J. Wang is with the School of Information and Communication Technology, Griffith University, Nathan, QLD 4111, Australia. E-mail: j.wang@griffith.edu.au.*

*Manuscript received 21 June 2016; revised 15 Mar. 2017; accepted 26 Mar. 2017. Date of publication 31 Mar. 2017; date of current version 5 July 2017.*
*Recommended for acceptance by L.B. Holder.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TKDE.2017.2690288*

targeted *greater Melbourne region*. For each sub-region, the selected influential users are able to successfully influence a certain percentage of users. By doing this, our proposed technique can help *Adidas-Australia* determine which set of $k$ influential users satisfies their needs, and then invite these users as boosters for the sporting event, or send sporting advertisements to the $k$ users, which can help attract a certain number of users in each sub-region along the planned running routes.

In these applications, the agents or companies not only need to target a certain number of social users, but also require the geographic influence spanning coverage of the selected users within the targeted regions to meet a predetermined minimum coverage criteria. To address this problem, we propose and investigate the problem of geo-social influence spanning maximization in location-aware social networks. Given a query with a query region $R$, a minimum covering percentage $\rho$ and a number $k$, we find the Maximal Geographical Spanning Regions (MGSR) $R' \subseteq R$ satisfying at least $\rho$ percent of users in $R'$ that can be influenced by $k$ users. Increasing $\rho$ results in smaller, more condensed spanning coverage, while decreasing $\rho$ results in larger, more sparse spanning coverage.

Although several other variations on the theme of influence maximization have been explored recently, previous work focused on the number of users to be influenced by the $k$ selected seed users. Aslay et al. [2] and Chen et al. [6] investigated topic-aware influence maximization based on the topic distributions within social networks, which finds the $k$-best seed users with the maximum influence spread in a social network using different topic-based influence propagation values. Li et al. [17] studied the problem of influence maximization in a constrained query region on location-aware social networks. Their goal was to select the $k$-best users in a social network which can influence the maximum number of social users in the given query region. None of the prior work considers the geographic spanning issue as presented here.

The significance of the MGSR problem is based on two key observations: (1) The users in the social network are rarely evenly distributed across a region. A small region may contain a large number of social users. It is also possible for a large region to contain a small number of social users. If every region contains the same number of users, the MGSR problem reduces to the problem of Li et al. [17]. But this assumption is often too strong to be practical in real applications; and (2) a social user may have a greater chance to influence neighbors in the physical world who are geographically close. To fill this research gap, we investigate the MGSR problem in this paper, and consider both the influence spread of seed nodes (expected number of activated nodes), and the geographic spanning area covered by the influence of seed nodes. The proposed MGSR problem and techniques described in this work enable users to more easily do geo-marketing and brand advertising.

Similar to all other work on influence maximization, the MGSR problem is also an NP-hard problem, which will be discussed in the following section. The addition of the minimum coverage $\rho$ increases the overall complexity of the problem since it is a bi-criteria optimization problem. To address the challenges of MGSR, we first show that the geographic influence spanning of a selected user can be modeled using a monotonic submodular function MGSR(). Then, we present a greedy approach to incrementally identify the top-$k$ initial users with a $1 - 1/e$ approximation ratio guarantee, and then develop an upper bound based approach to improve the efficiency. Although we can use a spatial index such as an R*-tree [4] or a QuadTree [13], [17] to prune the users outside of the query region, the greedy approach and the upper bound based approach are still expensive because they have to repeatedly compute and update the geographic influence span for each candidate user or the upper bound value. In order to further improve the efficiency, we develop an incremental algorithm using a new hybrid indexing structure which combines an R*-tree, and ordered influential user lists. We refer to this new index as an OIR$^*$-tree. The incremental algorithm consists of two main steps: First, the possible sub-regions within the given query region are checked to see if they are ($\rho$, $k$)-satisfactory; rather than probing users one by one as in the greedy approach; Next, the final $k$ nodes that can maximize the geographic influence span within the query region are identified. By doing this, we reduce the repeat of probing each user that has influence on the query region independently, other than probing the union of all possible $k$ users.

The main contributions of this work are:

- This is the first work to explore the problem of identifying the MGSR in a social network, which is significant and complementary research to the field of location-based social networks.
- We develop a new hybrid index—the OIR*-tree which combines the best features of ordered influential user lists and an R*-tree. The proposed index is suitable for use in a large number of spatial social influence maximization problems.
- One baseline approach and two advanced algorithms are proposed to efficiently compute MGSR.
- We evaluate the efficiency and effectiveness of the proposed techniques using three real datasets, and show the benefits of our methods when compared to previous related work [17].

The remainder of this paper is organized as follows. Section 2 presents the formalization of the MGSR problem. In Section 3, we present a greedy approach based on the submodular property of MGSR. Then, an upper bound based approach is developed to improve the efficiency in Section 4. To further reduce the computational cost, we then develop a novel index and propose an efficient indexing approach in Section 5. The results from the experimental study are discussed in Section 6. Finally, we review related work in Section 7, and conclude in Section 8.

## 2 The MGSR Problem

A location-aware social network is modeled as a directed graph $G = (V, E, P)$, where vertexes $V$ are users, edges $E$ are follower/followee relationships, and for any edge $u \rightarrow v$, $P(u, v)$ provides the propagation probability of $u$ to $v$. Each vertex $v \in V$ has a geographical location $(x, y)$ with longitude $x$ and latitude $y$. Initially, each vertex is inactive. If a vertex $u$ is selected as a seed, $u$ becomes active and will activate its out-neighbors. If $u's$ out-neighbor $v$ becomes active, $v$ will in turn activate $v's$ out-neighbors. There are two widely-adopted models—the independent cascade (IC) model and the linear threshold (LT) model [7], [14]. In this work, the IC model is adapted to compute the influence propagation probability between users. Each given seed node $u$, has a single chance to activate each inactive neighbor $v$ with an independent probability indicated by the edge weight $P(u,v)$. The newly activated node $v$ may further activate inactive neighbors. A node becomes active if and only if the propagation probability exceeds a certain threshold. This is a pragmatic adaptation of the IC model.

**Definition 1 (Geographic Grids GG).** *Consider a social network $G$ where each user has a geo-location. A two-dimensional geo-map of the social users in $G$ is partitioned into grids $\{GG_1, \ldots\}$. The grid size can be specified by the system as the total number of generated grids, or as the square distance of each grid (in kilometers for example).*

**Definition 2 (Reachable Grids of a Node).** *Let node $u$ be a seed of the activated node set $A(u)$ in a social network $G$. A grid $GG_i$ is reachable by the node $u$ if it contains at least one activated node in $A(u)$.*

However, reachable GGs may over-estimate the influence of a seed node in the geographic area based on Definition 2. This is because a geographic grid GG may contain many inactive nodes. To allow users to explicitly set constraints, we introduce an user-specified parameter $\rho$ to set the minimal covering ratio of a grid that can be selected for influence.

**Definition 3 (Countable GG of a Node).** *Given a node $u$ as a seed and the activated node set $A(u)$, a grid $GG_i$ is countable if and only if it is reachable by $u$, and it satisfies $\frac{|\{v|v \in GG_i \cap A(u)\}|}{|\{v|v \in GG_i\}|} \geq \rho$ where $|\{v|v \in GG_i\}|$ is the total number of nodes appearing in the coverage of grid $GG_i$, using the minimal covering ratio $\rho$.*

**Definition 4 (The Geographic Influence Span GIS of a Node).** *Given a node $u$ in a social network, the geographic influence span $\mathsf{GIS}(u)$ is the set of countable grids {GG} of the node $u$ based on Definition 3.*

Given a query region $R$, there may be many grids inside $R$. For a seed set $S$, if all of the grids inside $R$ are countable, then $R$ can be taken as the geographic influence span of $S$. But this is not true in most cases since the the distribution of social users is not geographically uniform. In this work, when a region is countable for a seed set $S$, then all grids inside the region must be countable for $S$. In addition, the number of nodes appearing in the countable regions is another important factor when evaluating the significance of the geographic influence span. Therefore, in this work we investigate the following problem: How can we select the $k$ vertices that maximize both the geographic influence span and the number of nodes to be covered in the span?

**Definition 5 (Maximum Geographic Spanning Region Query MGSR).** *Given a location-based social network $G = (V, E, P)$, a MGSR query $q$ with the parameters $(R, k, \rho)$, where $R$ is a query region, $k$ is the maximum number of seed nodes to be selected, and $\rho$ is the minimal covering ratio, find a $k$-vertex set $S \subseteq V$ and a set of influence spanning regions $\mathsf{GIS}(S) \subseteq R$ such that $\mathsf{MGSR}(S)$*

$$= \underset{S \subseteq V, |S| \leq k}{\operatorname{argmax}} \left\{ \alpha \times \frac{\{GG \in \mathsf{GIS}(S)\}}{\{GG \in R\}} + (1-\alpha) \times \frac{|\{v \in \mathsf{GIS}(S)\}|}{|V_R|} \right\}. \tag{1}$$

*subject to*

$$\frac{|\{v|v \in GG_i \cap A(S)\}|}{|\{v|v \in GG_i\}|} \geq \rho,$$

*for each grid $GG_i \in \mathsf{GIS}(S)$.*

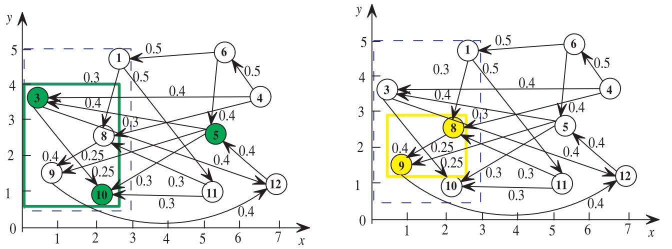


Fig. 1. A partial social network with edge influence.

Here, $\alpha$ balances the tradeoff between the geographic span and the number of nodes covered. $\{GG \in R\}$ represents the set of geographic grids inside the query region $R$. $\{GG \in \mathsf{GIS}(S)\}$ represents the set of geographic grids inside $R$ and they are countable for the seed set $S$. $|\{v \in \mathsf{GIS}(S)\}|$ represents the number of nodes appearing in the geographic grids of $\mathsf{GIS}(S)$ and $|V_R|$ is the total number of nodes inside the query region $R$. The parameters $\{GG \in R\}$ and $|V_R|$ are used for normalization.

From Definition 5, we can see that if $S$ is the selected as the set of $k$ seed nodes, then no other $k$-vertex set $S' \subseteq V$ satisfies $\mathsf{MGSR}(S') \geq \mathsf{MGSR}(S)$. When the parameter $\alpha = 0$, MGSR becomes the location-aware influence maximization problem [17], with the addition of a region constraint $\rho$. If the parameter $\alpha = 1$, then MGSR only considers the geographic influence span constrained by $\rho$. Therefore, in this work we present a definition of geographic influence spanning maximization in the general case. By setting a suitable $\alpha$ value (e.g., $\alpha = 0.5$), users can identify a $k$-vertex set, and find the corresponding best sub-regions in a targeted query region. In the above discussion, the fixed-size grids are used to describe the geographic span or coverage in order to make the definitions easy to understand. Alternatively, we calculate the exact geographic span or coverage in the examples and experiments by using the coordinates of social users.

**Example 1.** Consider the social graph in Fig. 1 and a query $q$ where the drawn box enclosed with the dashed lines represents a query region $R = \{(0, 0.5); (3, 5)\}$, $k = 1$, and $\rho = 50\%$. Assume that $\alpha = 0.5$ and if a node $u$ can successfully influence another node $v$ in the social graph, then its influential score to $v$ should be a certain value (say 0.3) based on the distribution of edge propagation probabilities [7], [14]. Taking the node $v_5$ as an illustration, it can influence the nodes $v_3$ (with 0.4), $v_{10}$ (with 0.3), $v_9$ (with 0.25), and $v_8$ (with 0.0). So, only two nodes ($v_3$ and $v_{10}$) have an influential score greater than or equal to 0.3, which are colored green. Their corresponding sub-region is marked by the green box. Given that two nodes are activated by $v_5$, the activation ratio in the green sub-region is $\frac{2}{4} = 50\%$, which satisfies the minimal covering ratio $\rho = 50\%$. Therefore, the green sub-region is a satisfied influence spanning region $\{(0, 0.5); (2.5, 4)\}$. As such, $\mathsf{MGSR}(v_5) = 0.5 \times \frac{(2.5-0)\times(4-0.5)}{(3-0)\times(5-0.5)} + (1-0.5) \times \frac{4}{5} = 0.724$. Similarly, if we select node $v_8$ (not $v_5$) as a seed node, then only the node $v_9$ can be influenced. Since $v_8$ is located inside of the query region, the yellow sub-region consisting of the two nodes is a countable region $\{(0.5, 1.5); (2, 2.5)\}$. Thus, $\mathsf{MGSR}(v_8) = 0.5 \times \frac{(2-0.5)\times(2.5-1.5)}{(3-0)\times(5-0.5)} + (1-0.5)\times \frac{2}{5} = 0.255$. If $\rho$ is increased up to 70 percent, the green sub-region cannot be treated as the influenced region because the minimal covering ratio does not hold for any single node. So $v_5$ cannot be the seed candidate, but $v_8$ still does. If $\rho$ is increased up to 70 percent and $k$ is set as 2, then the two-best seeds would be $\{v_5, v_1\}$ and $\mathsf{MGSR}(\{v_5, v_1\}) = 1$. In this example, we assume that the green (or yellow) sub-region is the geographic grid. Otherwise, the geographic grids inside the green (or yellow) sub-region must be recursively accessed.

**Theorem 1 (Duplicate Avoidance).** *Given any two nodes $u_1$ and $u_2$, if both nodes appear in the top-k seed set S, then there are no duplicates when computing the score of MGSR(S).*

In simple terms, if node $u_1$ (or $u_2$) can individually influence a geographic region successfully, then the successive node $u_2$ (or $u_1$) can not influence the region again when calculating the overall score MGSR(S) for the seed set $S$. Theorem 1 satisfies the targeted expectation in our problem—MGSR selecting a $k$-vertex set that can influence a geographical spanning area maximally. It also makes the comparison of MGSR possible across different candidate seed sets due to the following property.

**Property 1 (Order Insensitive MGSR Computation).** *The MGSR computation of the optimal seed set is insensitive to the selection order of the seed nodes.*

**Proof.** Based on Equation (1), we can easily see that MGSR(S) is order-insensitive if $GIS(S)$ is order-insensitive because the other parameters are invariant in the equation. Consider two nodes $u_1$, $u_2$ in $S$. According to Definitions 1, 2, 3, and 4, we have $GIS(S)$ = $\{\mathrm{GG}\}_{u_1}$ + $\{\mathrm{GG}\}_{u_2}$ − $\{\mathrm{GG}\}_{u_1} \cap \{\mathrm{GG}\}_{u_2}$ + $\{\mathrm{GG}\}_{u_1 u_2}$ where $\{\mathrm{GG}\}_{u_1}$ represents the set of countable geographic grids for $u_1$, and $\{\mathrm{GG}\}_{u_1 u_2}$ represents the set of geographic grids that are countable for $u_1$ and $u_2$ together, but not for individual node $u_1$ or $u_2$. Therefore, $GIS(S)$ can be rewritten as $GIS(u_1) + GIS(u_2) - GIS(u_1) \cap GIS(u_2) + \{\mathrm{GG}\}_{u_1 u_2}$. From this transformation, we can see that $GIS(S)$ is order-insensitive if $S$ contains two nodes. When $S$ has more than two nodes, the above procedure is also applied by considering $u_1$ and $S \setminus \{u_1\}$, concluding the proof. □

Reconsider the nodes $v_5$ and $v_1$ in Fig. 1 as an example of knowing the optimal seed set $S$ consisting of the two nodes. If we first select $v_5$ and then choose $v_1$ in $S$, the second seed $v_1$ can only contribute to a small area covering itself, because the area bounding the nodes $v_3$, $v_8$, $v_9$ and $v_{10}$ has been covered by seed node $v_5$. If $v_1$ is selected as the first seed, then it will identify a larger area covering $v_1$ and $v_8$. When $v_5$ is selected as the second seed into $S$, the remaining spanning region will be discovered. Both strategies generate the same spanning region as the final result.

**Property 2.** MGSR *is NP-hard.*

The MGSR problem can easily be proven to be NP-hard by reduction from the influence maximization problem [14], and computing the exact location-aware influence span can be shown to be NP-hard by reduction from the influence spread problem [7]. The only requirement is to set $\rho = 0$ and $V_R = V$, where $V_R$ is the node set covered by the query region.

**Property 3.** MGSR $(\cdot)$ *is a monotone submodular function.*

**Proof.** If the following inequality holds (1) MGSR(.) is a monotonic function. $\mathsf{MGSR}(S) \leq \mathsf{MGSR}(T)$ for any node sets $S \subseteq T$; (2) $\mathsf{MGSR}(S \cup \{u\}) - \mathsf{MGSR}(S) \geq \mathsf{MGSR}(T \cup \{u\}) - \mathsf{MGSR}(T)$ for all nodes $u$ and any node sets $S \subseteq T$, then the function is a monotone submodular function.

Since $S \subseteq T$, $S$ must have $A(S) \subseteq A(T)$ where $A(S)$ and $A(T)$ represent the node sets activated by the seed node sets $S$ and $T$ respectively. Therefore, we have $\mathsf{MBR}(A(S)) \leq \mathsf{MBR}(A(T))$. According to Definition 4, the geographic influence spanning value is calculated from the MBR of the activated node set. So we can derive $\mathsf{MGSR}(S) \leq \mathsf{MGSR}(T)$, which means that MGSR(.) is a monotonic function.

Since MGSR(.) is a monotonic function, the geographic influence spanning area of $T$ must cover the geographic influence spanning area of $S$. When a new seed node $u$ is added into the corresponding node sets $S$ or $T$, the nodes activated by $u$ appear in the influence spanning area of $S$ must also appear in the influence spanning area of $T$, but not vice versa. So, $u$ may activate more nodes outside of the influence span of $S$ than that of $T$. Because of the node uniqueness constraint in Property 1, $\mathsf{MGSR}(S \cup \{u\}) - \mathsf{MGSR}(S) \geq \mathsf{MGSR}(T \cup \{u\}) - \mathsf{MGSR}(T)$ always holds for all nodes $u$, and any node sets $S \subseteq T$. □

## 3 Greedy Approach

The submodular property of MGSR allows the seed node to be incrementally selected with the maximum marginal gain in influence for a query region. Then, the valid geographic influence span is identified based on the activated nodes in the query region. Finally, the search is incrementally expanded until $k$ verified seed nodes are found, or until the geographic influence span of the current seed node set cannot be increased any further. The approximate MGSR approach can achieve a $(1 - 1/e)$ approximation ratio by greedily identifying the top-$k$ seeds. Our approach uses incremental expansion with a submodular set function subject to a knapsack constraint as originally described by [20] in Theorem 1.

To do this, the incoming and outgoing influence for any node to a given node $v$ is pre-computed. This produces two indexes—an incoming list $L_{in}(v)$ and an outgoing list $L_{out}(v)$ for each node $v$ in the graph. To reduce the index size without loss of quality, nodes with an influence less than a certain small value are pruned from the index.

In Algorithm 1, we first load all of the candidate nodes using the pre-computed incoming and outgoing indexes. Then, the algorithm runs for $k$ iterations to select $k$ nodes with the maximal geographic spanning region value. In each iteration, the candidate node set is probed to find the best node that has the maximum marginal gain, i.e., a node $u$ maximizing $\mathsf{MGSR}(S \cup \{u\}) - \mathsf{MGSR}(S)$, where $\mathsf{MGSR}(S \cup \{u\})$ and $\mathsf{MGSR}(S)$ are calculated using Definition 3. After $k$ iterations, the $k$ best nodes that have been found are returned.

**Algorithm 1.** MGSR Greedy Approach

```
Input: A graph $G = (V, E)$, a user query $q = (R, k, \rho)$.
Output: $S$—the $k$-vertex set.
Find the nodes $V_R$ appearing in $R$ based on R*-tree index.
Initialize the candidate hash table $C \leftarrow \phi$.
for each node $v \in V_R$ do
    Load the incoming list $L_{in}(v)$.
    for each node $u \in L_{in}(v)$ do
        Store $C(u) \leftarrow C(u) \cup \{v\}$.
for $i \leftarrow 1$ to $k$ do
    $\Delta \leftarrow 0$
    for each node $u \in C \setminus S$ do
        if $\Delta < MGSR(S \cup \{u\}) - MGSR(S)$ then
            $selection \leftarrow u$
            $\Delta \leftarrow MGSR(S \cup \{u\}) - MGSR(S)$
    $S \leftarrow S \cup selection$
return $S$
```

## 4 Upper Bound Pruning Approach

To improve the efficiency of the greedy algorithm, we now develop an upper bound based approach to address the MGSR problem, which can prune out unnecessary candidate nodes during the computation. Before getting into the details of the upper bound based algorithm, we first present a lemma to show the existence of the upper bound.

**Lemma 1.** *For any node $u$, the* MGSR *marginal gain at $1:k$ iterations cannot exceed the* MGSR *value when $u$ is the sole seed in $S$.*

Since $\mathsf{MGSR}(.)$ was proven to be a monotone and submodular function in Property 3, we can show that $\mathsf{MGSR}(S_{i-1} \cup \{u\}) - \mathsf{MGSR}(S_{i-1}) \geq \mathsf{MGSR}(S_i \cup \{u\}) - \mathsf{MGSR}(S_i)$ always holds where $S_{i-1} \subseteq S_i$. That is, the maximal gain when taking a node as a new seed in the early steps of the process must be larger than or equal to that of taking the node in the late steps. For any node, the maximal gain is not increased by the number of iterations. As such, we can see that $\mathsf{MGSR}(u)$ is an upper bound on the maximal gain produced by $u$ for any $i$th iteration ($i \in [1:k]$).

Given a node and an exact maximal gain value at the $i$th iteration, Lemma 1 can be applied to safely skip the nodes with an upper bound value less than the node's true maximal gain value. Furthermore, the upper bound values provide a probing priority for the nodes in the algorithm. That is, the nodes with highest upper bound values will be accessed first. For a given query region, all of the nodes $V_R$ appearing in the query region $R$ are retrieved, and the influential nodes are retrieved from the index $L_{in}(v)$ for each node $v \in V_R$. By aggregating the accessed nodes, the complete seed candidate set $S = V_R \cup \{L_{in}(v) | v \in V_R\}$ can be found. To incrementally select and verify all $k$ seed nodes with a maximum marginal gain, the seed candidates are sorted by the upper bound value of the geographic influence span within the query region $R$, where the regional upper bound of the MGSR of a node is defined in Definition 6.

**Definition 6 (Regional Upper Bound of a Node).** *Assume a set of nodes $A_{S_{i-1}}$ have been activated by $S_{i-1}$ at the $(i-1)$th moment. Consider the node $u$ being the $i$th seed where $0 \leq i \leq k$. Given a geographic region $R$, and $L_{out}(u) \cap V_R \neq \phi$, the upper bound of the geographic influence span of $u$ in $R$ is defined as the countable geographic grids $GG$ over the nodes in $\{L_{out}(u) \cap V_R\}$ and $\{v \in A_{S_{i-1}} | v \in GG(L_{out}(u) \cap V_R)\}$.*

However, the regional upper bound of a node may have overlaps with previous seeds. So it cannot be directly used as the upper bound. To address this, the incremental portion must be computed as the actual countable upper bound, and is the "new reward" effect of introducing the node as a new seed.

**Definition 7 (Countable Regional Upper Bound of a Node).** *The countable regional upper bound of a node is the regional upper bound minus the overlaps with the previous seeds.*

Based on the countable regional upper bound of a node $u$ and the number of nodes covered by the countable MBR, the incremental score of MGSR can be computed as $\mathsf{MGSR}(S_{i-1} \cup \{u\}) - \mathsf{MGSR}(S_{i-1})$.

**Example 2.** Consider the example in Fig. 1. Assume the node $v_5$ is in the seed set at the current moment and the influence span is enclosed in the green box. When the node $v_1$ is probed, the regional upper bound value is the whole query region in the box {(0, 0.5); (3, 5)} where $v_3$ and $v_{10}$ have been activated by $v_5$; $v_8$ and $v_9$ are included in the influential region of $v_5$; $v_1$ is the only one node to be newly activated. Since the overlapping area $\{(0, 0.5); (2.5, 4)\}$ to be influenced by $v_5$ must be excluded, the residual is the countable regional upper bound value of $v_1$ which is $(3-0) \times (5-0.5) - (2.5-0) \times (4-0.5) = 13.5 - 8.75 = 4.75$. Therefore, $\mathsf{MGSR}(\{v_5, v_1\}) - \mathsf{MGSR}(\{v_5\}) = 0.5 \times \frac{4.75}{3 \times 4.5} + 0.5 \times \frac{1}{5} = 0.28$.

The upper bounding procedure is presented in Algorithm 2. We first load the candidate set $C$ for the query $q$. Then, the upper bound value for each candidate node in $C$ is initialized. These candidate nodes are maintained in a heap $M$ where the candidate nodes are ordered in a descending order of upper bound values. After that, $k$ iterations are ran to find the $k$ best candidates. In each iteration, we always probe the candidate node $u'$ with the maximum upper bound and calculate the marginal MGSR gain $\Delta(u')$. If the marginal MGSR gain $\Delta(u')$ is larger than or equal to the upper bound value for the next node, then $u'$ can be safely selected as the best choice in this iteration. Otherwise, the computed marginal MGSR gain $\Delta(u')$ and $u'$ will be re-added back to the heap $M$, where $u'$ in $M$ may be reconsidered in a late iteration. If possible, $u'$ is selected as the best choice without any further computation since the current MGSR value of $u'$ is an exact value, and the value is the largest one in $M$. The procedure is presented in Lines 8-19.

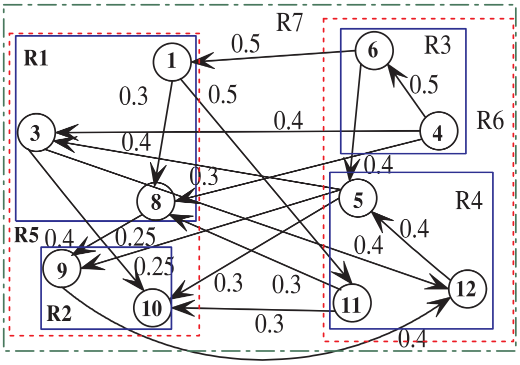


Fig. 2. R*-tree representation of a social graph.

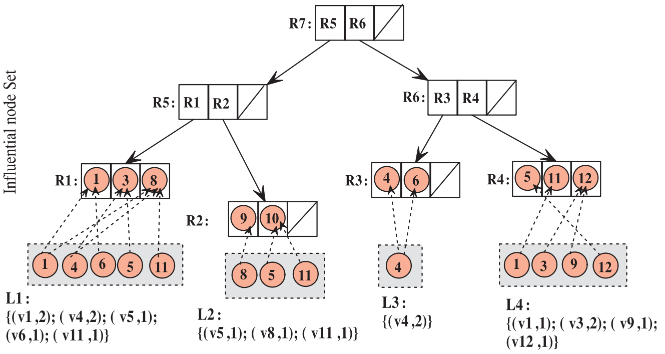


Fig. 3. OIR*-tree data structure of a social graph.

**Algorithm 2.** MGSR Upper Bound Approach

```
Input: A graph $G = (V, E)$, a user query $q = (R, k, \rho)$.
Output: $S$—the $k$-vertex set.
Find the nodes $V_R$ appearing in $R$ based on R*-Tree index.
Load the candidate node set $C$ using Lines 3-6 in Algorithm 1.
Initialize a sorted heap $M \leftarrow \phi$.
for each node $u \in C$ do
    Compute $MGSR(u)$ and record $u \rightarrow MGSR(u)$ into $M$.
for $i \leftarrow 1$ to $k$ do
    $Stop \leftarrow$ false
    while not $Stop$ do
        Get the top node $u'$ and the $MGSR$ value $\Delta(u')$ from $M$.
        if $u'$ has not been visited then
            $\Delta(u') \leftarrow MGSR(S_{i-1} \cup \{u'\}) - MGSR(S_{i-1})$
            if $\Delta(u') \geq MGSR(u_{next})$ of the second node $u_{next} \in M$ then
                $S \leftarrow S \cup \{u'\}; C \leftarrow C \setminus \{u'\}$;
                $i \leftarrow i + 1; Stop \leftarrow$ true;
            else
                Move $u' \rightarrow \Delta(u')$ to the right position in $M$ based on the updated bound $\Delta(u')$.
        else
            $S \leftarrow S \cup \{u'\}; C \leftarrow C \setminus \{u'\}$;
            $i \leftarrow i + 1; Stop \leftarrow$ true;
return $S$
```

## 5 The OIR*-Tree Index Based Solution

Although the upper bound pruning approach can improve the efficiency of the greedy approach, the computational cost is still high because the upper bound is not tight. In both the greedy approach and the upper bound approach, it is difficult to improve the efficiency since the MGSR computation can be repeated. Another limitation in the two approaches is the effectiveness. Even though the submodular property guarantees the algorithms can achieve a $(1 - 1/e)$-precision, there is inevitably some loss in overall effectiveness. Since we know the selection of the seed nodes depends on previously selected seed nodes in the $k$ iterations, we now propose a heuristic approach to remove the dependencies in seed node selection.

The basic idea is that given a query region, first find all of the sub-regions to be covered by the query region using an OIR$^*$-tree based on a hybrid social-and-spatial index. Then, incrementally determine if the sub-region satisfies the given query requirement $(\rho, k)$. If so, a $k$-vertex set that can activate $\rho$ percent of the nodes in a sub-region exists. By doing this, a subset of significant sub-regions are identified. Starting from the significant sub-regions with a maximum span, we can compute the MGSR and find the corresponding candidate seed nodes. Since the significant sub-regions are identified by their own $k$-vertex sets, the aggregated $k$-vertex sets may be larger than the maximal allowable value $k$. To discover the final $k$-vertex set, select and expand every potential sub-region that has been identified until the geographic influence span cannot be expanded any further.

### 5.1 Identifying ($\rho$, $k$)-Satisfactory Sub-Regions

In this section, our aim is to find all of the sub-regions where $\rho$ percentage of nodes can be activated using a $k$-sized vertex set. By doing this, a set of significant sub-regions can be identified as the components of the MGSR candidates, and the sub-regions can be filtered without requiring any unnecessary MGSR computations.

Each sub-region is assessed based on the incoming node lists ($L_{in}(v)$) of the nodes in the sub-region. The nodes in the sub-regions and their respective incoming node lists can be maintained using a combination of an R*-tree and ordered influential node lists-producing a hybrid index that we will refer to as an OIR$^*$-tree index henceforth. A similar hybrid index was proposed by [10] and [18], where an R-tree and inverted files were combined for spatial keyword search.

In this paper, we develop a new data structure to support spatial influence analysis in social networks. It is a hybrid index for influential nodes. For each vertex stored as an OIR$^*$-tree node, the corresponding incoming node list is computed and stored as an ordered influential node list. The nodes in the list are ranked by "activation power" in the current sub-region. The activation power of a node in a sub-region is approximated by the number of nodes in the sub-region that can be activated by the node. In addition, the intermediate results of the sub-regions at the lowest level in the tree can be reused to check the ($\rho$, $k$)-satisfaction of a sub-region at higher levels. Figs. 2 and 3 present an example of the indexing structure used for maintaining the social graph in Fig. 1.

**Example 3.** Consider a query ($R_5$, 0.7, 2) on the data in Fig. 3 where $\alpha = 0.5$. Since there are two sub-regions $R_1$ and $R_2$ at the leaf level, we will first check to see if they are ($\rho$, $k$)-satisfied. To do this, the influential node set is accessed in the order of nodes in $L_1$. Since $\rho$ is set as 0.7, there are three choices to activate $R_1$ – {$v_1$, $v_4$}, {$v_1$, $v_5$}, or {$v_4$, $v_6$}. For $R_2$, there are only two choices – {$v_5$, $v_8$} or {$v_8$, $v_{11}$}. In this case, both are ($\rho$, $k$)-satisfied. To process internal MBR nodes such as $R_5$, we need to dynamically compute the ordered influential node list based on the child MBR nodes. In this example, this is $L_5$: $\{(v_1, 2);\ (v_4, 2);\ (v_5, 2);\ (v_{11}, 2);\ (v_6, 1);\ (v_8, 1)\}$, which aggregates $L_1$ and $L_2$ together. Since $\rho$ is set as 0.7 and $k = 2$, we need to check if we can find at most two seed nodes that can successfully activate $\lceil 0.7 \times 5 \rceil = 4$. From the ordered list $L_5$, we can see there are four candidate nodes since each of them can activate two nodes. After checking the activated nodes, we find that {$v_1$, $v_5$} would be the result for $R_5$.

From the above example, we can terminate our candidate search when identifying $R_5$ using the node $v_{11}$ in $L_5$. In $L_5$, we know the best selection of a seed node can activate at most two nodes. Based on the requirement – $\lceil 0.7 \times 5 \rceil = 4$, selecting the second seed node requires at least two more nodes to be activated. Therefore, the candidate search can be stopped by $v_{11}$ in $L_5$ because each of the remaining nodes can only activate one node.

Algorithm 3 provides the checking procedure for the sub-regions covered by a query. For each sub-region, check to see if the $k$ seed nodes can activate $\rho$ percent of nodes in the sub-region. To do this, a bottom-up strategy as shown in Lines 4-23 is used. If the current sub-region is a leaf node in the OIR$^*$-tree, then check if $\rho$ percent of the nodes can be activated by at most $k$ seed nodes, as shown in Lines 5-14. The function PROCESS_LEAFNODE in Algorithm 4 performs the computation for the sub-regions at the leaf node, while the function RESULTREUSE in Algorithm 5 is applied to promote the intermediate results of a sub-region to a parent node. For any sub-region at the internal level, if all of the child nodes have been processed, then the node is checked to see if it is ($\rho$, $k$)-satisfied based on the intermediate results of the node's children, as shown in Lines 16-23. The exact approach of how to process the sub-regions in the internal level is presented in Algorithm 6.

In Algorithm 4, we show the procedure of processing the sub-regions at the leaf level. When the algorithm is initialized, a list of ordered influential nodes is obtained. For each influential node, the nodes influenced are selected. Then, the influential node in the first position of the list is taken as a seed node in each iteration, and the current percentage of nodes that can be activated is computed. After each influential node is processed, the order of the remaining influential node list is updated by removing the activated nodes. The iteration continues until $\rho$ or $k$ is satisfied, as shown in Lines 5-18. In Line 9, *u.value* is the number of influenced nodes which is recorded in the ordered influential node list $L_{R_i}$. In Lines 14-16, we remove the nodes that can be activated by $u$ because active nodes cannot be activated again. In Line 17, we resort the influential node list $L_{R_i}$ based on the removal of the active nodes. Finally, two computed results are returned. If a result exists (*judged* = **true**), then $S_{temp}$ is the minimal number of seed nodes that can activate $\geq \rho$ percent of the nodes. Otherwise, $S_{temp}$ is a $k$ node set that can be used as an intermediate result when computing the sub-regions of the parent nodes.

**Algorithm 3.** Sub-Region Checking Algorithm

**Input:** A location-aware social graph $G = (V, E)$ maintained in OIR$^*$-tree $T$, and a user query $q = (R, \rho, k)$.

**Output:** An updated OIR$^*$-tree with a ($\rho$, $k$)-satisfied sub-region marked.

```
1:  Find all the sub-regions $R_{set} = \{R_0, R_1, \ldots\}$ covered by the
    query region $R$ using the OIR*-Tree index.
2:  // Process sub-regions bottom-up.
3:  while $R_{set} \neq \phi$ do
4:    $R_i \leftarrow$ POP($R_{set}$)
5:    if $R_i$ is a leaf node then
6:      (judged, $S_{temp}$) ← PROCESS_LEAFNODE($R_i$)
7:      if judged = true then
8:        Mark $R_i$ as ($\rho$, $k$)-satisfied sub-region in OIR*-tree.
9:      else
10:       Mark $R_i$ as non-($\rho$, $k$)-satisfied sub-region in
          OIR*-tree.
11:     if $R_j \in R_{set}$ covering $R_i \neq \varnothing$ then
12:       // Promote intermediate result of $R_i$ to $R_j$.
13:       RESULTREUSE($R_i$, $R_j$)
14:       Disable $R_i$ as a child sub-region of $R_j$.
15:   else
16:     if all child sub-regions of $R_i$ have been processed then
17:       (judged, $S_{temp}$) ← PROCESS_INTERNALNODE($R_i$)
18:       if judged = true then
19:         Mark $R_i$ as ($\rho$, $k$)-satisfied sub-region in
            OIR*-tree.
20:       else
21:         Mark $R_i$ as non-($\rho$, $k$)-satisfied sub-region in
            OIR*-tree.
22:     else
23:       Delay $R_i$ until the child sub-regions are processed.
24: return Updated OIR*-tree with a ($\rho$, $k$)-satisfied
    sub-region marked.
```

**Algorithm 4.** PROCESS_LEAFNODE($R_i$)

```
1:  Retrieve the ordered influential node list $L_{R_i}$ for $R_i$.
2:  Initialize judged ← false
3:  Initialize $p \leftarrow 0$
4:  Initialize temporary set $S_{temp} \leftarrow \phi$
5:  while !judged and $p \leq |L_{R_i}|$ and $|S_{temp}| < k$ do
6:    Set vertex $u \leftarrow L_{R_i}[p]$
7:    Add $u$ into $S_{temp}$
8:    // $u$.value is the number of nodes $u$ can influence in the
      current subregion $R_i$.
9:    activated_num += $u$.value
10:   if activated_num $\geq \rho \times V_{R_i}$ then
11:     judged ← true
12:   else
13:     $U \leftarrow U \cup A_i(u)$ where $A_i(u)$ is the influenced node set
        in $R_i$ for the influential node $u$.
14:     for all $u' \in L_{R_i}$ do
15:       $A_i(u') \leftarrow A_i(u') \setminus A_i(u)$
16:       $L_{R_i}[u'] \leftarrow |A_i(u')|$
17:     Sort $L_{R_i}$.
18:   Increment $p$.
19: return judged and $S_{temp}$
```

Algorithm 5 shows how the intermediate results are maintained such that they can be reused to test for ($\rho$, $k$)-satisfaction of the sub-regions of parent nodes. If a probed sub-region is the first child of a parent node, then the information is recorded, as shown in Lines 3-6. Otherwise, the activation power for each of the influential nodes is computed on the fly to generate an aggregated influential node list, as shown in Lines 8-14. In Line 8, we aggregate the two influential node lists via a union operation, where for a node appearing in the two lists, the values of the node are added together, and the aggregated value of the node is stored in the aggregated list. Here, a duplicate count can occur since an influential node may influence the same node in different subregions. The duplicate counts can be accounted for when the influenced nodes are promoted to $V_{R_j}$ in Line 10-14. In particular, Lines 14 adjusts the duplicate count. By doing this, the full influential node list can be obtained for the sub-region at the parent level.

**Algorithm 5.** ResultReuse($R_i$, $R_j$)

```
1:  // Here, $R_j$ is the parent MBR node of $R_i$.
2:  Assume $V_k^i$ are the $k$ nodes selected to satisfy ($\rho$, $k$) in
      sub-region $R_i$, and $L_{R_i}$ is the ordered node list of
      nodes influencing $R_i$.
3:  if $L_{R_j} = \varnothing$ then
4:      $V_{R_j} \leftarrow V_{R_i}$
5:      $V_k^{pool} \leftarrow V_k^i$
6:      $L_{R_j} \leftarrow L_{R_i}$
7:  else
8:      $L_{R_j} \leftarrow$ aggregate $L_{R_j}$ and $L_{R_i}$.
9:      $V_k^{pool} \leftarrow$ aggregate $V_k^{pool}$ and $V_k^i$.
10:     for each node $v \in V_{R_i}$ do
11:         if $v \notin V_{R_j}$ then
12:             Insert $v$ into $V_{R_j}$.
13:         else
14:             Update $L_{R_j}$ by minus 1 from the value of each
                  influential node $u$ where $v \in A(u)$ and $u \in L_{R_j}$.
15: return An updated sub-region $R_j$.
```

Algorithm 6 is used to validate the ($\rho$, $k$)-satisfaction of sub-regions of the internal nodes based on the intermediate obtained from their child nodes. Different from Algorithm 4, Algorithm 6 only needs to access a subset of influential nodes in $V_k^{pool}$ and the corresponding influenced node sets. This is because the subset of influential nodes are large enough to be used to work out the new $k$-vertex set for the internal nodes. The correction can be guaranteed using Property 4. At the beginning of the algorithm, we need to access all node lists $L_{R_j}$ and get all of the influential nodes that have a higher ranked positions than the position of the last node in $V_k^{pool}$. Here, $V_k^{pool}$ maintains a set of influential nodes, and each influential node has an influenced node set for region $R_j$. The influential nodes in $V_k^{pool}$ are sorted based on the number of influenced nodes. The length of $V_k^{pool}$ may be larger than $k$, and $R_j$ is one of the child nodes of $R_i$ in the OIR $*$hbox$-$tree. By doing this, we can get all of the necessary influential nodes and corresponding influenced node sets for the region $R_j$ by accessing the child sub-regions $R_i$. In the *While-Loop*, the maximum number of nodes to be influenced by a $k$-vertex set is computed. Based on the intermediate results, the region $R_i$ is checked to see if it satisfies the ($\rho$, $k$) conditions or not. If it is true, $R_i$ is marked as a countable MBR.

Given a sub-region, there may be a large number of nodes that can influence the sub-region. In practice, only a few nodes with high influence can get into the top-$k$ set. To reduce the number of low scoring nodes to be probed, we propose the following heuristic pruning rule.

**Algorithm 6.** Process_InternalNode($R_j$)

```
Assume $L_{R_j}$ maintains a set of ordered node-value pairs,
  and $V_k^{pool}$ maintains a set of ordered node-nodes pairs.
Find the position $x$ in $L_{R_j}$ for the last node of $V_k^{pool}$.
Find the subset $L'_{R_j}$ from $0 \rightsquigarrow x$ positions over $L_{R_j}$.
Initialize judged $\leftarrow$ false.
Initialize $p \leftarrow 0$
Initialize a temporary set $S_{temp} \leftarrow \phi$
while !judged and $p \leq |L_{R_j}|$ and $|S_{temp}| < k$ do
    Get the vertex $u$ at the pointer position in $L'_{R_j}$.
    if $u \notin V_k^{pool}$ or $L'_{R_j}[u] > |V_k^{pool}[u]|$ then
        Get the influenced nodes of $u$ from previously
          processed sub-regions.
    $S_{temp} \leftarrow S_{temp} \cup u$
    $activated\_num \leftarrow activated\_num + u.value$
    if $activated\_num \geq \rho \times V_{R_i}$ then
        judged $\leftarrow$ true
    else
        $U \leftarrow U \cup V_k^{pool}[u]$
        Update $L'_{R_j}$ by removing the active nodes in $V_k^{pool}[u]$.
    Increment $p$.
return judged and $S_{temp}$
```

**Property 4 (Pruning of $k$ Node Selection Pool).** *Consider two small sub-regions $R_i$ and $R_j$ with $V_k^i$ and $V_k^j$ as the respective $k$ node sets with the maximal influence, with corresponding ordered influential node lists denoted as $L_{R_i}$ and $L_{R_j}$. Assume there is a large sub-region $R_N$ that can cover $R_i$ and $R_j$. To work out the seed set $V_k^N$ of $R_N$, we only need to explore partial nodes from the influential node lists $L_{R_i}$ and $L_{R_j}$. For $L_{R_i}$ or $L_{R_j}$, we only need to explore the nodes with influence higher than the $\epsilon$ times of $v_{lowest}$'s influence where $v_{lowest}$ is the node with the lowest influence in $\{V_k^i \cup V_k^j\}$.*

The proof of Property 4 comes from the following intuition. Consider a vertex $v_x$ that is one of $k$ nodes for $R_N$ and the node's influence is much smaller than that of vertex $v_{lowest}$ in the node list $L_{R_N}$. Note that the list of nodes has been sorted by the influence of nodes to the region. So we know the position of a node in $L_{R_N}$ represents the influential power on the region. In this context, the assumption of selecting $v_x$ in a $k$ vertex set of $R_N$, rather than $v_{lowest}$, means that $v_x$ has more influence to $R_N$ than $v_{lowest}$. If $v_x$ has much less influence than $v_{lowest}$ in all small sub-regions ($R_i$ and $R_j$), then it will have small likelihood to be in top-$k$ candidate set for the large sub-region ($R_N$). As such, this pruning strategy allows us to focus on the significant nodes with a high likelihood. Therefore, we use the lowest position of a node $v_{lowest} \in \{V_k^i \cup V_k^j\}$ to bound the discovery of the $k$ vertex set with regards to $R_N$, without building the full ordered influential node list $L_{R_N}$.

### 5.2 Finding Best $k$ Seeds

At this stage, all sub-regions have been checked and labeled. The marked results are maintained in the updated OIR$^*$-tree. To find the final $k$ vertex set that maximizes the

TABLE 1
Statistical Information for the Datasets

| Datasets | #Vertexes | #Edges | AvgD | MaxID | MaxOD |
|---|---|---|---|---|---|
| Gowalla | 197 K | 1.9 M | 9.67 | 739 | 735 |
| Twitter | 554k | 4.29 M | 7.75 | 1,143 | 639 |
| Foursquare | 4.9 M | 53.7 M | 11.6 | 4,702 | 727 |

MGSR within the query region, each sub-region candidate is probed, and the coverage across horizontal regions is expanded since different sub-regions may share seed nodes. In particular, when $k$ is large, sub-regions can be activated by a small number (less than $k$) of nodes. In these cases, the sub-regions can be combined together by merging the vertex sets to achieve a maximum MGSR score. The straightforward solution is to make pair-wise comparisons for any two sub-regions, but this is expensive. To reduce the computational cost, two approaches are explored: Exhaustive expansion and an $\epsilon$-approximate expansion.

*Exhaustive Expansion*: The key idea of exhaustive expansion is to probe the sub-regions level by level using the updated OIR$^*$-tree. As shown in Algorithm 3, the sub-region of a parent node in the updated OIR$^*$-tree covers all of the sub-regions of the child nodes in the same tree based on the geographic coverage. Therefore, there are several conditions that can be used to induce early termination:

- If a sub-region can be successfully selected as a result, then all sub-regions beneath can be skipped;
- Given two countable sub-regions $R_i$ and $R_j$ in the tree, if the selected $k$ sets $V_k^i$ and $V_k^j$ are disjoint, $V_k^i \cap V_k^j = \phi$, then the other sub-regions in the subtree rooted at $R_j$ do not need to be probed;
- If $V_k^i \supseteq V_k^j$, then $R_j$ and $R_i$ can be merged and the other sub-regions in the subtree rooted at $R_j$ do not need to be probed;
- Assume that the updated OIR $*$hbox$-$tree has been probed up to level $l_{current}$. At this moment, the intermediate results for some of the expanded sub-regions have been calculated. Based on the sub-regions, and the number of nodes covered by the sub-regions, the MGSR score is computed using the intermediate results. The calculated MGSR score is now the lower bound value of each sub-region candidate. The upper bound value of each sub-region candidate at the level $l_{current}$ is also computed. Here, the upper bound value depends on the addition of the current node, and the other sub-regions overlapping with the sub-region in $l_{current}$. For certain sub-region candidates, if the upper bound values are lower than the lower bound of a sub-region expansion candidate, then the candidate can be skipped.

By doing this, the final $k$ node set and the maximum geographic influence spanning coverage can be obtained. At some point, only one sub-region expansion candidate is left, which can be returned as the best expansion. The remaining task is to expand only the best one by probing the other possible sub-regions that have not been checked, and cannot be covered by the selected sub-regions in the best expansion.

*$\epsilon$-Approximate Expansion*: Although the exhaustive expansion algorithm can terminate early, it still has to first probe all possible combinations, and then make the final decision based on the MGSR values of the combinations. This tion addresses how to terminate the expansion algorithm as early as possible while bounding the accuracy with a predetermined approximation ratio.

TABLE 2
Statistical Properties of the Test Queries

| **Gowalla** | **Q1** | **Q2** | **Q3** |
|---|---|---|---|
| **#Regional Nodes** | $8k$ | $9k$ | $11k$ |
| **#Seed Candidates** | $28k$ | $35k$ | $43k$ |
| **Twitter** | **Q1** | **Q2** | **Q3** |
| **#Regional Nodes** | $57k$ | $59k$ | $63k$ |
| **#Seed Candidates** | $307k$ | $315k$ | $338k$ |
| **Foursquare** | **Q1** | **Q2** | **Q3** |
| **#Regional Nodes** | $314k$ | $278k$ | $349k$ |
| **#Seed Candidates** | $2.1m$ | $1.9m$ | $2.4m$ |

The main idea of $\epsilon$-approximate expansion is to terminate the expansion of a candidate by checking if the lower bound is equal to or larger than $\epsilon\times$ the upper bound. If this is true, then the expansion for the candidate can be terminated. At the same time, all of the optimizations in the exhaustive expansion algorithm can also be exploited. After all of the expansions have terminated, the whole expansion algorithm can be safely stopped with the accuracy bounded by the $\epsilon$ approximation. By doing this, the algorithm can achieve early termination without probing the sub-regions at the lowest levels. As such, the computational cost can be further reduced.

## 6 Experimental Study

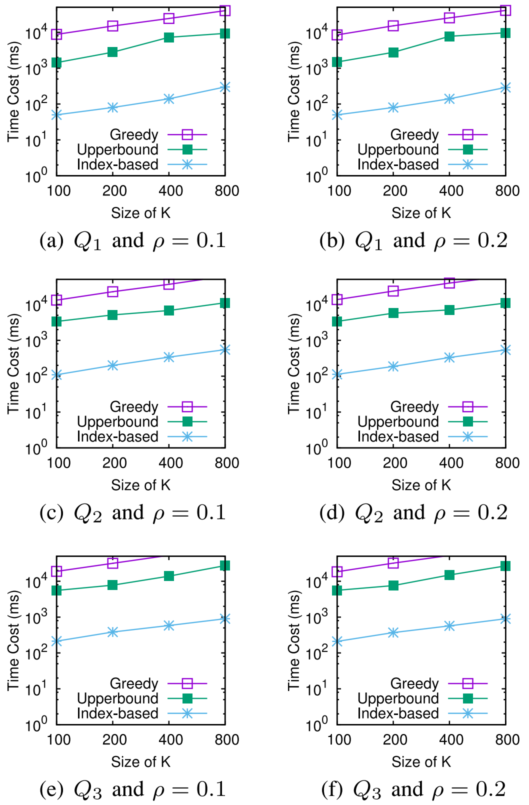
### 6.1 Experimental Settings

*Algorithms.* We study the performance of the following algorithms: the greedy algorithm (*Greedy*) from Section 3, the upper bound approach (*Upperbound*) from Section 4, and the OIR*-tree index solution with an $\epsilon$-approximate expansion (*Index-based*) from Section 5.

*Test Datasets.* We use three real datasets *Gowalla*, *Twitter*, and *Foursquare*. All three datasets were downloaded from the author's website[1] [17]. The user location is the place the user most frequently checked in. The three datasets are directed graphs and the details are shown in Table 1, where AvgD denotes the average degree, and MaxID/MaxOD denotes the maximum in-/out-degree.

*Test Queries.* We generated three types of queries with different regional nodes, denoted as $Q_1$, $Q_2$ and $Q_3$. The symbol of $Q_{11}$ is used to show the experimental result when running $Q_1$ over the 1st dataset *Gowalla*. Similarly, $Q_{22}$ represents $Q_2$ ran over the 2nd dataset *Tweet*, and $Q_{31}$ represents $Q_1$ ran over the 3rd dataset *Foursquare*. We briefly use $Q_1$, rather than $Q_{11}$ when we discuss the experimental results for the *Gowalla* data. Each query type is comprised of 50 queries with similar sizes of regional nodes, and we report the mean performance. In Table 2, we show the approximate sizes of regional nodes and the sizes of potential seed candidates for different queries and different datasets. All algorithms were implemented in C++ 10.0 on

[1] http://dbgroup.cs.tsinghua.edu.cn/ligl/laim/

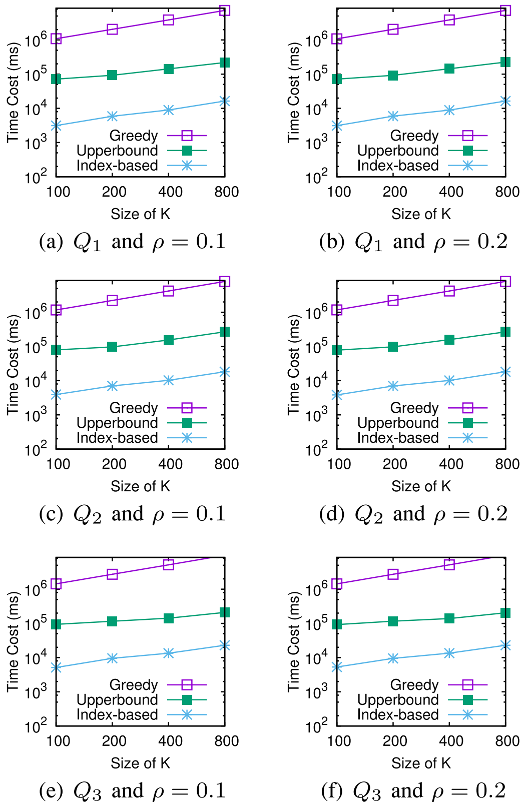


(a) $Q_1$ and $\rho = 0.1$ (b) $Q_1$ and $\rho = 0.2$

(c) $Q_2$ and $\rho = 0.1$ (d) $Q_2$ and $\rho = 0.2$

(e) $Q_3$ and $\rho = 0.1$ (f) $Q_3$ and $\rho = 0.2$

Fig. 4. Running time for queries on the Gowalla.



(a) $Q_1$ and $\rho = 0.1$ (b) $Q_1$ and $\rho = 0.2$

(c) $Q_2$ and $\rho = 0.1$ (d) $Q_2$ and $\rho = 0.2$

(e) $Q_3$ and $\rho = 0.1$ (f) $Q_3$ and $\rho = 0.2$

Fig. 5. Running time for queries on the Tweet.

Windows 7, and run on an Intel(R) i5 CPU @2.60 GHz with 8 GB RAM.

### 6.2 Efficiency Evaluation

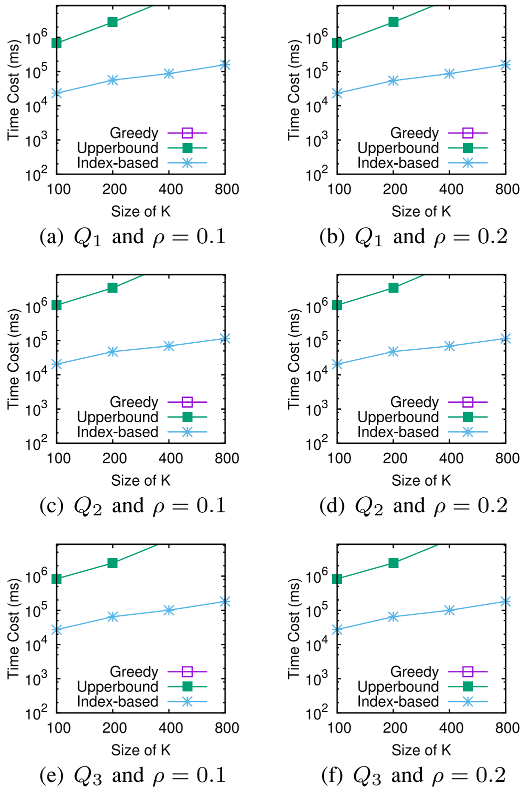
In this section, we evaluate the efficiency of the proposed approaches over different datasets with different parameter settings. Here, we vary the size of $k$ from 100 to 800. And $\rho$ is selected as 0.1 and 0.2 since all three datasets are sparse. If we give a high value to $\rho$, the run often results in an empty result set for the specified sizes of $k$ between 100 and 800. In addition, we run three types of queries over each dataset and report the mean time performance.

*Gowalla Dataset*: From Fig. 4, we can see that our proposed index-based approach performs much better than the greedy and upper bound approaches on the Gowalla dataset. In addition, the experimental results also show that the time cost of the three algorithms grows linearly as $k$ increases, and varying $\rho$ values does not affect the running time.

Taking $Q_1$ as an example with $\rho = 0.1$. When $k = 100$, the greedy algorithm and the upper bound approach take $8{,}759$ and $1{,}429$ ms, respectively. Our index-based algorithm takes only 50 ms. In this case, the upper bound approach improved the greedy algorithm 6 fold, while the index-based approach has a 175x improvement. When $k = 800$, the greedy algorithm and the upper bound based approach take $40{,}120$ and $9{,}345$ ms, respectively. Our index-based algorithm requires only 300 ms. In this case, the upper bound approach improved the greedy algorithm by 4x, while the index-based approach had a 133x improvement. From this study, we see that the index-based approach is significantly better than the other approaches.

When we increase the number of regional nodes and seed candidates as $Q_2$ and $Q_3$, the three algorithms take much longer to identify the $k$ best seeds. As shown in Figs. 4a, 4c, and 4e, the greedy algorithm requires $15{,}005$, $22{,}685$ ms, and $3{,}1667$ ms for $Q_1$, $Q_2$, and $Q_3$ when $k = 200$. With the same configuration, the upper bound approach requires $2{,}793$, $5{,}111$, and $7{,}770$ ms, but our index-based approach needs only 80, 200, and 383 ms. As $k$ increases, the greedy and upper bound approaches perform even worse. However, the index-based approach is much more scalable. The main reason is that the greedy algorithm repeatedly scans the complete seed candidate list, and the upper bound based approach also needs to repeatedly probe a large part of seed candidate list, but the index-based approach only needs to do a one-pass, local $(\rho, k)$ verification.

*Tweet Dataset:* Fig. 5 shows the time cost of the three algorithms on the *Tweet* dataset. The trend of time cost grows linearly with the increase of $k$ from 100 to 800. For $Q_1$ shown in Fig. 5a, the greedy algorithm takes about 16, 34, 64, and 121 minutes when $k$ is set as 100, 200, 400 and 800. The upper bound based approach takes about 1.1, 1.5, 2.3, and 3.6 minutes with different $k$ values. However, our index-based approach requires only 3, 5, 8, and 16 seconds for the same tasks. From the experimental study, we conclude that our index-based approach outperforms the other two algorithms, although all scale linearly with respect to $k$. Varying $\rho$ has no observable impact on the running time of the three algorithms. When we increase the number of regional nodes and seed candidates as shown in Figs. 5a, 5c, and 5e, the running times for all three algorithms increase significantly. In the case of Fig. 5e ($k = 800$), the greedy algorithm needs 2.7 hours to evaluate the query $Q_3$. The upper bound based



(a) $Q_1$ and $\rho = 0.1$ (b) $Q_1$ and $\rho = 0.2$

(c) $Q_2$ and $\rho = 0.1$ (d) $Q_2$ and $\rho = 0.2$

(e) $Q_3$ and $\rho = 0.1$ (f) $Q_3$ and $\rho = 0.2$

Fig. 6. Running time for queries on the foursquare.

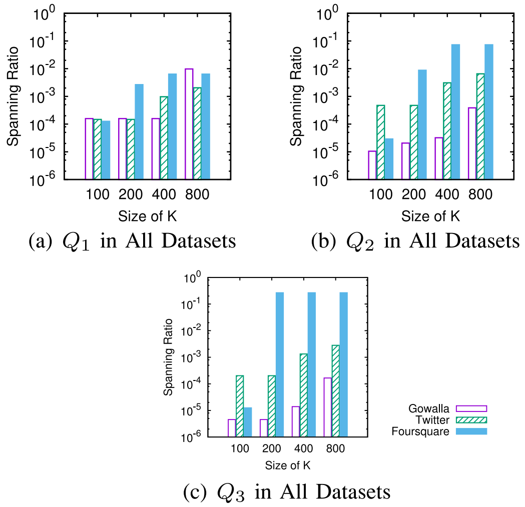
approach requires 3.4 minutes, and the index-based approach can finish the task in 23 seconds.

*Foursquare Dataset*: Fig. 6 shows the performance of the three algorithms on the *Foursquare* dataset. Our index-based approach greatly outperforms the other two algorithms. When $k = 100$, the greedy algorithm takes 13 hours for $Q_1$, 18 hours for $Q_2$, and 22 hours for $Q_3$ over the Foursquare dataset. For the three queries, the upper bound based approach takes about 10, 14, and 18, but the index-based approach only needs about 20, 24, and 27. In Fig. 6, we have only shown the experimental results that completed in less than 2.3 hours.

### 6.3 Effectiveness Evaluation

We selected three specific regions with longitudes and latitudes to demonstrate the effectiveness of the approaches. The queries were $Q_1$ = {10.822811 −122.30276; 37.856213 −100.29089}, $Q_2$ = {30.262996 −122.30276; 37.856213 −97.750338} and $Q_3$ = {20.262996 −122.30276 37.856213 −97.750338}. We evaluate the three cases as the same regional queries over different datasets. Here, we only use the geo-region size as a metric, i.e., the overall region size of $Q_1$ is calculated as (37.856213−10.822811) × (122.30276−100.29089) = 595.0557. Similarly, we can get the other two region sizes. To illustrate the effectiveness, we measure our proposed geo-social influence spanning model by using the three metrics including the influence spanning coverage, the varied trend of the influence spanning ratio, and the spatial reachability in geography.

Fig. 7 shows the spanning ratio for the influence coverage of the selected $k$ best seeds for different queries and datasets. From the experimental results, we can see that the influence spanning ratio may not change as the size of seed nodes is increased. For example, $k$ is 100, 200, 400 for Gowalla in Fig. 7a, $k$ is 100 and 200 for Twitter in Figs. 7a, 7b, and 7c, and $k$ is 200-800 for Foursquare in Fig. 7c. Since the social users are not distributed evenly on geo-map based locations, there is no observable effect on the ratio. In addition, we observe that the approaches get better results on the Foursquare dataset, than on the Twitter dataset or the Gowalla dataset. For example, selecting 200 seeds on Foursquare can influence about 26.7 percent the region of $Q_3$, but only 0.02 percent for Twitter and 0.0004538 percent for Gowalla. This study coincides with real properties of the investigated datasets. The Foursquare collection contains more social users in the investigated region for $Q_3$ (11k nodes for Gowalla, 63k nodes for Twitter, and 349k nodes for Foursquare).

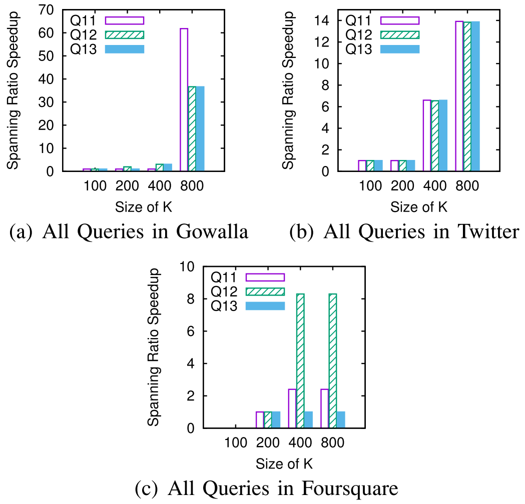


(a) $Q_1$ in All Datasets (b) $Q_2$ in All Datasets

(c) $Q_3$ in All Datasets

Fig. 7. Influence spanning coverage comparison.

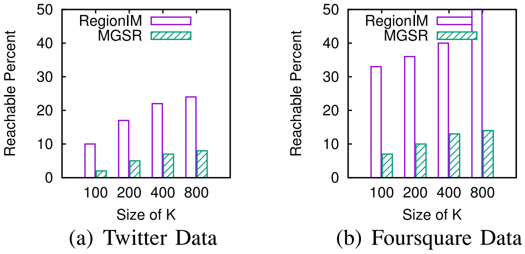
Fig. 8 shows the speedup for spanning ratios when we increase the number of seed nodes for the same dataset. From the results of Gowalla in Fig. 8a, a big increase in performance happens when $k$ is 800 where the speedup metric uses the spanning coverage of $k = 100$ as a base. This means that 800 is a good starting point for us to specify the parameter $k$ if we want to influence a certain sub-region on the Gowalla social network. This is because all three queries



(a) All Queries in Gowalla (b) All Queries in Twitter

(c) All Queries in Foursquare

Fig. 8. Influence spanning speedup comparison.



(a) Twitter Data (b) Foursquare Data

Fig. 9. Reachable evaluation of activated nodes on $Q_1$.

have large performance gains when $k$ is 800, but not for 200 and 400, when compared with $k = 100$. Similarly, we can see that 400 is a good value for the Twitter collection in Fig. 8b. For Foursquare, shown in Fig. 8c, we use the spanning coverage of $k = 200$ as the base of the speedup metric because the coverage of $k = 100$ is too low. But no clear good starting points can be found because there is a large gap in the speedup values for the three queries for both 400 and 800. From these results, we can see that the spanning coverage ratio does not always increase when $k$ increases. The results also verify that our proposed MGSR model can successfully obtain the maximum coverage for different types of geographic distributions in different datasets.

In order to show the effect of the seed nodes to be selected by [17] (denoted as *RegionIM*) and our proposed model *MGSR*, we developed a novel metric to measure the geographic reachability of activated nodes. Simply speaking, we assign a radius to each node where the radius is the average geographic distance from the node to its social neighbors. Given a region query, *RegionIM* and *MGSR* can both identify seed node sets and activated node sets. The goal of the new metric is to evaluate the percentage of the pairs of nodes in the activated node set which are reachable within the nodes' radius. For example, given two nodes $v_1$, $v_2$, they are reachable if and only if the geographic distance between $v_1$ and $v_2$ is not larger than the sum of their radii. A lower percentage means the activated nodes can influence a larger region. So, these nodes have a better spatial distribution.

We take $Q_1$ as an example to show the difference of the two methods. Here, $\rho$ was set to 0.2 and $\alpha$ was set to 0.5 in the evaluation. Fig. 9 shows the experimental results for the two large datasets Twitter and Foursqure. On Twitter, *RegionIM* recommended the top-$k$ seeds. The activated nodes are pairwise reachable from 10 to 24 percent when $k$ varies from 100 to 800, based on the computed radii. But *MGSR* can reduce the reachable ratio to 2-8 percent. On the Foursquare collection, the activated nodes in *RegionIM* reached each other from 33 to 50 percent, but *MGSR* can reduce the reachability from 7 to 14 percent when $k$ varies. From these results, we can see that the geographic distribution of activated nodes in *MGSR* is better than that of *RegionIM*.

### 6.4 Space Cost Evaluation

In this section, we present the space usage when using an OIR$^*$-tree index to evaluate the three types of queries over the three test collections where the original data size is 27.7 MB for *Gowalla*, 149.7 MB for *Tweet*, and 1.58 GB for *Foursquare*. As shown in Table 3, the space cost of building the OIRT$^*$-tree index is 19.1, 105.0, and 989.6 MB for the three datasets, respectively. We measure the space cost of evaluating the three types of queries in the datasets using the proposed algorithms. Since we load the whole index into memory before starting query evaluation, the consumed memory is at least the size of the index for each dataset. From the experimental results, we can see that the space cost is similar for the same query over a dataset. For example, $Q_1$, $Q_2$ and $Q_3$ consumed about 22.6–23.5 MB for *Gowalla*, 114.7–115.7 MB for *Tweet*, and 1.08–1.11 GB for *Foursquare*.

TABLE 3
Space Consumption for the Three Datasets

| Space Cost | Index (MB) | $Q_1$ (MB) | $Q_2$ (MB) | $Q_3$ (MB) |
|---|---|---|---|---|
| Gowalla | 19.1 | 22.6 | 22.7 | 23.5 |
| Tweet | 105 | 115.5 | 114.7 | 115.7 |
| Foursquare | 989.6 | 1,105 | 1,084 | 1,106.9 |

## 7 Related Work

The influence maximization problem was originally proposed by Domigos and Richardson [11]. The two proposed methods are probabilistic, and the influence spread was not bounded. Kempe et al. [14] proposed two discrete influence spread models, the Independent Cascade model and the Linear Thresholds model. They provided proof that the influence maximization problem can be solved using a greedy algorithm with a $1 - \frac{1}{e}$ approximation ratio for both models. Since the influence maximization problem is NP-hard, there are many studies investigating alternative approaches to improve the efficiency. Kimura and Saito [15] used shortest paths to estimate the IC model. Leskovec et al. [16] developed a "lazy-forward" algorithm which preformed much better than simple greedy algorithms. Chen et al. [7] proposed the PMIA algorithm to solve the influence spread maximization problem using the IC model. The main idea was to estimate the global influence of vertex $v$ based on the local maximum influence in-arborescence (PMIIA), which is a tree structure representing the union of maximum influence paths from other vertexes to $v$. A similar idea was applied to support the LT model [8].

Influence maximization has been investigated in many other forms. For example, the problem of topic-aware influence maximization has received considerable attention recently [2], [3], [6], [9]. Since each edge of any two users in the social network may be weighted differently for different topics, the different weights on edges results in a different selection of the $k$ seed users based on the comparison of the general influence maximization definition. Li et al. [17] studied the influence maximization problem in a given query region. Feng et al. [12] proposed a method to solve the influence maximization using a novelty decay model. Time constrained influence maximization has also been studied [19]. In addition, [21] considered the social influence maximization using a diversification constraint.

All the above work focused the maximum number of nodes to be influenced by the $k$ selected seeds as a metric for studying the problem of influence maximization. In contrast, the proposed geo-social influence spanning maximization problem can allow search users to get the $k$ best seed nodes and see the corresponding maximum geographic influence spanning for the specified query regions.

## 8 Conclusion

In this paper, we propose and formally define the novel problem of maximum geographic spanning regions over location-aware social networks, which takes a query region, $k$ seeds, and a locally minimal covering ratio $\rho$ as parameters. Our approach can compute the top-$k$ selected seed nodes, and capture locality effects. By doing this, query users can easily observe the quality of the selected $k$ seeds based on the geographical coverage within the query region. The larger the geographical coverage influenced by the $k$ seeds, the better the quality of the seed selection strategy. Using the approach, users can easily determine the seed set that maximally influences the users' preferred regions, providing a new and convenient way to make decision on when and where to launch marketing campaigns.

To address the computational challenges, we presented a greedy solution and a upper bound pruning approach to incrementally identify the most influential seed nodes. We also designed an OIR$^*$-tree index and traversal algorithm to accelerate the computation using a bottom-up strategy. The experimental results verified the performance of our proposed index and approaches in terms of efficiency, effectiveness and space cost. Besides geographic influence maximization problems, the investigated index and algorithms in this work can also be applied to spatial-social data analytical problems such as spatial-social community detection.

## Acknowledgments

This work was mainly supported by the ARC Discovery Projects under Grant No. DP140101587 and DP160102114. This research is also partially supported by the ARC Discovery Projects under Grant No. DP130103051, DP160102412 and DP170104747. The authors are thankful for partial support from NSFC (No. 61370080 and No. 61170007) and the Shanghai Innovation Action Project (No. 16DZ1100200). Jianxin Li and Zhenying He are the corresponding authors in this work.

## References

[1] N. Anstead and B. O'Loughlin, "Social media analysis and public opinion: The 2010 UK general election," *J. Comput.-Mediated Commun.*, vol. 20, no. 2, pp. 204–220, 2015.

[2] Ç. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates, "Online topic-aware influence maximization queries," in *Proc. 17th Int. Conf. Extending Database Technol.*, 2014, pp. 295–306.

[3] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 81–90.

[4] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R*-Tree: An efficient and robust access method for points and rectangles," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1990, pp. 322–331.

[5] A. Bruns and T. Highfield, "Political networks on twitter: Tweeting the queensland state election," *Inform. Commun. Soc.*, vol. 16, no. 5, pp. 667–691, 2013.

[6] S. Chen, J. Fan, G. Li, J. Feng, K. Tan, and J. Tang, "Online topic-aware influence maximization," *Proc. VLDB Endowment*, vol. 8, no. 6, pp. 666–677, 2015.

[7] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1029–1038.

[8] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 88–97.

[9] W. Chen, T. Lin, and C. Yang, "Real-time topic-aware influence maximization using preprocessing," *CSoNet*, pp. 1–13, 2015.

[10] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial Web objects," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 337–348, 2009.

[11] P. M. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2001, pp. 57–66.

[12] S. Feng, X. Chen, G. Cong, Y. Zeng, Y. M. Chee, and Y. Xiang, "Influence maximization with novelty decay in social networks," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 37–43.

[13] R. A. Finkel and J. L. Bentley, "Quad trees: A data structure for retrieval on composite keys," *Acta Inf.*, vol. 4, pp. 1–9, 1974.

[14] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 137–146.

[15] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," in *Proc. 10th Eur. Conf. Principle Practice Knowl. Discovery Databases*, 2006, pp. 259–271.

[16] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 420–429.

[17] G. Li, S. Chen, J. Feng, K. Tan, and W. Li, "Efficient location-aware influence maximization," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 87–98.

[18] Z. Li, K. C. K. Lee, B. Zheng, W. Lee, D. L. Lee, and X. Wang, "IR-tree: An efficient index for geographic document search," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 4, pp. 585–599, Apr. 2011.

[19] B. Liu, G. Cong, D. Xu, and Y. Zeng, "Time constrained influence maximization in social networks," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 439–448.

[20] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Res. Lett.*, vol. 32, no. 1, pp. 41–43, 2004.

[21] F. Tang, Q. Liu, H. Zhu, E. Chen, and F. Zhu, "Diversified social influence maximization," in *Proc. IEEE/ACM Int. Conf. Advances Social Netw. Anal. Mining*, 2014, pp. 455–459.

[22] C. B. Williams and G. J. Gulati, "Social networks in political campaigns: Facebook and the congressional elections of 2006 and 2008," *New Media Soc.*, vol. 15, no. 1, pp. 52–71, 2013.

**Jianxin Li** received received the PhD degree in computer science from the Swinburne University of Technology, Australia, in 2009. He is a senior lecturer in the School of Computer Science and Software Engineering, University of Western Australia. His research interests include database query processing & optimization, social network analytics, and traffic network data processing.

**Timos Sellis** received the PhD degree in computer science from the University of California, Berkeley, in 1986. He is a professor with the Swinburne University of Technology, Australia. Till the end of 2012, he was the director of the Institute for the Management of Information Systems (IMIS) and a professor with the National Technical University of Athens, Greece. Between 2013 and 2015, he was a professor with RMIT University, Australia. His research interests include big data, data streams, personalization, data integration, and spatio-temporal database systems. He is a fellow of the IEEE and the ACM.

**J. Shane Culpepper** received the PhD degree from the University of Melbourne, in 2008. Since then, he has been a faculty member with RMIT University, with research interests include designing efficient algorithms and data structures for a wide variety of information storage and retrieval problems.

**Zhenying He** received the BS, MS, and PhD degrees in computer science from the Harbin Institute of Technology, China, in 1998, 2000, and 2006, respectively. Currently, he is an associate professor in the School of Computer Science, Fudan University. His current research interests include keywords search on structured data, query processing on RDF data, and big data.

**Chengfei Liu** received the BS, MS, and PhD degrees in computer science from Nanjing University, China, in 1983, 1985, and 1988, respectively. Currently, he is a professor in the Faculty of Science, Engineering and Technology, Swinburne University of Technology. His current research interests include keywords search on structured data, query processing and refinement for advanced database applications, query processing on uncertain data and big data, and data-centric workflows. He is a member of the IEEE and the ACM.

**Junhu Wang** received the PhD degree in computer science from Griffith University, Australia, in 2003. He is currently an associate professor in the School of Information and Communication Technology, Griffith University. His research interests include query processing, integrity constraint reasoning, and graph algorithms.