Target-Aware Holistic Influence Maximization in Spatial Social Networks

Taotao Cai, Jianxin Li, Ajmal Mian, Rong-Hua Li, Timos Sellis, *Fellow, IEEE*, and Jeffrey Xu Yu

Abstract—Influence maximization has recently received significant attention for scheduling online campaigns or advertisements on social network platforms. However, most studies only focus on user influence via cyber interactions while ignoring their physical interactions which are also essential to gauge influence propagation. Additionally, targeted campaigns or advertisements have not received sufficient attention. To address these issues, we first devise a novel holistic influence diffusion model that takes into account both cyber and physical user interactions in an effective and practical way. Based on the new diffusion model, we formulate a new problem of *holistic influence maximization*, denoted as *HIM* query, for targeted advertisements in a spatial social network. The *HIM* query problem aims to find a minimum set of users whose holistic influence can cover all target users in the network, which belongs to a set covering problem. Since the *HIM* query problem is NP-hard, we develop a greedy baseline algorithm and then improve on this algorithm to reduce the computational cost. To deal with large networks, we also design a spatial-social index to maintain the social, spatial and textual information of users, as well as developing an index-based efficient solution. Finally, we conduct extensive experiments using one synthetic and three real-world datasets to validate the efficiency and effectiveness of the proposed holistic influence diffusion model and our developed algorithms.

Index Terms—Holistic influence maximization, targeted advertisements, spatial social networks

1 Introduction

Recently, the increasing number of online social media users has resulted in an invaluable opportunity to exploit the spread of product adoption, ideas, and news through social networks. Research along this direction has gained significant attention. For instance, some researchers have focused on the diffusion process [1], [22], the prediction of future diffusion [6], learning the strength of user-to-user influence [8], and the influence spread estimation [21]. Another line of research concentrates on the problem of *influence maximization* (*IM*) [10] that aims at selecting k influential users such that they will cause the maximum influence spread in a social network. Following this line of work, variants of the *IM* problem have been investigated recently, such as *topic-aware IM* [2], [4], *targeted IM* [18], [24], *time-constrained IM* [19], *location-aware IM* [16], [27], [30], and *community-aware IM* [17]. However, all these works only studied the cyber interactions of users in analyzing their social influence on each other. Although the *IM* problem has been studied in location-based social networks [16] such methods do not count for the opportunity of users to influence each other through physical interactions. Thus, these methods may miss the additional influence spread in assessing a user's influence in the selection process of seed users. Moreover, location is one of the critical factors to evaluate user-to-user's influence in the *IM* problem, e.g., the region-aware based *IM* problem was studied in [12] and the venue-aware based *IM* problem was studied in [30].

To meet the requirement of targeted advertisements, the problem of *targeted IM* has been studied in [18], [24]. Both works aim to find a set of seed users whose social influence can cover the maximum number of "target" users. Differently, Li *et al.* [18] define the target users w.r.t. a keyword query, i.e., the users mention some keywords in the query. A weighted reverse influence set sampling technique is developed to address the keyword based *targeted IM* problem. Song *et al.* [24] define the targeted influence spread that is constrained to an event location and a deadline of running the event, i.e., the users are targeted if they have a probability to check-in at this event location before the deadline. However, none of these methods investigate how users propagate their influence to other users via their spatial interactions together with social influence. Motivated by [11], we argue that a user may have a certain probability to influence his/her *spatially-close* users with similar interests. However, it may not be practical to assess user-to-user spatial influence impact only with spatial distance and interests. A user in a crowded region may have many neighbors with similar interests, and it cannot be assumed that each of these neighbors is equally influenced because one generally contacts with a limited number of persons around him/her in daily life.

To address the above issues, we first develop a holistic influence spread model by considering three important factors - *social connection*, *spatial connection*, and *preference-based*

- *T. Cai and J. Li are with the Deakin University, Geelong, Victoria, VIC 3220, Australia. E-mail: {taotao.cai, jianxin.li}@deakin.edu.au.*
- *A. Mian is with the University of Western Australia, Perth, WA 6009, Australia. E-mail: ajmal.mian@uwa.edu.au.*
- *R.-H. Li is with the Beijing Institute of Technology, Beijing 100811, China. E-mail: lironghuascut@gmail.com.*
- *T. Sellis is with the Swinburne University of technology, Melbourne, VIC 3122, Australia. E-mail: tsellis@swin.edu.au.*
- *J.X. Yu is with the Chinese University of Hong Kong, Hong Kong, China. E-mail: yu@se.cuhk.edu.hk.*

Manuscript received 9 Apr. 2019; revised 2 June 2020; accepted 11 June 2020. Date of publication 17 June 2020; date of current version 7 Mar. 2022. (Corresponding author: Jianxin Li.) Recommended for acceptance by W. Zhang. Digital Object Identifier no. 10.1109/TKDE.2020.3003047

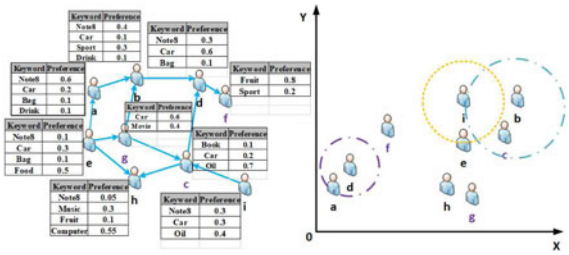


Fig. 1. An example of *HIM* (holistic influence maximization) query.

similarity connection together, by which the practical influence of users can be analyzed in various applications. Based on the proposed holistic influence spread model, we formulate a novel *holistic influence maximization* problem, called as *HIM* query problem, for targeted advertisements in spatial social networks. Consider a social network $G = (V, E, T, L)$ where V is a set of users, E is a set of cyber links of users in V, T is a feature or interest space of users in V, and L maintains the location information for each user in V. Given a targeted advertisement expressed by a keyword query Q, the *HIM* query problem is to find the minimum set of seed users whose holistic influence spread can cover all the target users with regards to Q in G. Unlike the traditional IM problem, the *HIM* query problem belongs to the *set covering* problem because it requires the seed user set to influence all the target users rather than the maximum number of users.

Example 1 (Motivation). Fig. 1 provides an example to show a study about the *HIM* query problem. Consider the crisis communication plan of *Samsung company* for their faulty product "Note8" smartphone. *Samsung* needs to issue a total recall campaign on their sold *Note8* due to a potential hazard issue. Meanwhile, they also want to reduce the negative impact of this product in marketing. Generally, smartphone companies do not have full records of customers. A possible way for *Samsung* is to select a set of seed users from social networks and motivate them to share the recall information by giving out some rewards. Meanwhile, by sharing social messages or talking to the target users around them, the selected seed users may be persuaded to propagate the positive and energetic remedies of *Samsung* with regards to the recall event. From a business perspective, *Samsung* certainly hopes to select the minimum number of seed users that can cover all the target users by running this campaign – Noted that the target users in this scenario are represented by the users who may purchase the *Note8* smartphone in the social networks.

In this example, $\{a, b, d, e, h, i\}$ are the target users w.r.t. *Note8*. If the social influence diffusion like [24] is applied, then the optimal seed set would be $\{e, i\}$ in order to activate all the target users. If the spatial influence diffusion is additionally considered, then the sole seed {i} is enough to activate all the target users because {i} can activate {e} via its spatial influence. Thus, for the HIM query problem, the optimal seed set would be $\{i\}$.

Besides, the *HIM* query may bring more benefits to different real applications. In *Event Recommendation Campaigns*, the influential users selected by a *HIM* query can make their influenced users to meet up those who are spatially close to some others. In *Social Link Analysis*, the *HIM* query can help to uncover the hidden social links of users by tracking users' social influence and spatial influence. There are other applications *HIM* query may advance, e.g., users' behavior analytics using cyber or physical activities, engaging spatial-social community detections, and news/ads propagation type analytics.

Several challenges exist in solving the *HIM* query problem. First, it is non-trivial as it is even harder than the conventional *IM* problem which is an NP-hard problem itself [10]. In *HIM* query problem, we need to estimate the social influence spread of users as well as compute their spatial influence spread. Second, the spatial influence of users cannot be materialized in advance because the assessment is based on users' spatial distance and their interests w.r.t. the online keyword query. In other words, the user-to-user spatial influence is dynamic rather than static. Third, few previous studies consider multiple factors in addressing *IM* or *targeted IM* problem. Thus, it is not possible to directly apply the existing algorithms or index to solve the *HIM* query problem.

To address the above challenges, in this paper, we first develop a holistic influence spread model which is a dynamic metric consisting of three factors w.r.t. online keyword queries. Since the computation of spatial influence spread is one of the main bottlenecks in solving the *HIM* query problem, we then develop two algorithms based on different search strategies, by which we can reduce a large amount of unnecessary computation for those users who are unlikely to become the eligible seed users. We also devise a novel and simple spatial-social index R_{SS} to maintain the structural, location information and interests of users, which can support the upper bound based pruning. Finally, we develop a R_{SS} index based algorithm to solve the *HIM* query problem efficiently. Our contributions are summarized as follows:

1) We propose a new holistic influence diffusion model by considering users' social connections, spatial connections, and preference-based similarity connections together in a holistic manner.
2) We propose a *HIM* query that complements the classic *IM* problem in real applications, and verify that it is a NP-hard problem with the proved approximation ratio.
3) We develop one baseline algorithm by extending the *WRIS* method in [18] and two efficient algorithms to answer *HIM* queries.
4) We design a novel but simple spatial-social index to effectively maintain users' information as well as support efficient pruning in *HIM* query processing. We also show the procedure of our novel index maintenance.
5) We verify the effectiveness of our proposed holistic diffusion model and the efficiency of our developed algorithms using four datasets.

We present the preliminaries in Section 2 and formalize the *HIM* query in Section 3. Then, we develop three algorithms, a novel index and the maintenance of the index in Section 4.1. Experimental results are reported and discussed in Section 5. After that, we discuss the related works and

the different focus of this work in Section 6. Finally, this paper is concluded in Section 7.

2 Preliminaries

In this section, we review the targeted influence maximization problem as well as the solutions [18].

2.1 Keyword Based Targeted Influence Maximization

Consider a social network $G = (V, E, T)$ where each vertex u in V is a social user in G, each edge $e = (u, v)$ in E represents a cyber link or a social relationship between u and v, and T is a keyword space to represent user's interests. We use a weighted term vector $T_v = \{T_v^1, T_v^2, \ldots\}$ to represent the user v's interests (*keywords*) and the weight of each interest, and T_v^i ($i \in [1, |T_v|]$) is represented as $T_v^i = \{keyword(i) : weight(i)\}$, e.g., $T_a^1 = \{Note8 : 0.6\}$. Moreover, for each user v, we collect his/her interests by analyzing his/her posted messages and learn the weight of each interest using the standard *tf-idf* model with normalized *idf* vector.

Given an advertisement or campaign expressed by a keyword query Q, the impact of the advertisement or campaign on a user u is defined as

$$\phi(u, Q) = \sum_{w \in Q} tf_{w,u}.idf_w. \tag{1}$$

Thus, the influence spread of the targeted advertisement can be calculated by accumulating the impact on each user that can be influenced by the selected influential user set. Here, the influential user set is denoted as S, and their influenced user set is denoted as $I(S)$. Let $I^Q(S)$ denote the influence score of $I(S)$ w.r.t. the keyword query Q. Then, the expected influence score can be calculated using

$$E[I^Q(S)] = E\left[\sum_{v \in I(S)} \phi(v, Q)\right] = \sum_{v \in V} p(S \mapsto v) \cdot \phi(v, Q). \tag{2}$$

Here, $p(S \mapsto v)$ is the probability with which a user v is activated by the influential user set S.

Thus, the target of *the keyword based influence maximization* (KB-TIM) problem is to find a set S of seed users satisfying $OPT_k^Q = max_{S \subseteq V, |S|=k} E[I^Q(S)]$.

2.2 Influence Diffusion Model

In [18], under the *independent cascade* (*IC*) model, the *influence probability* $p(e)$ of an edge $e = (u, v)$ is used to measure the social impact from user u to v. Every user is either in an *activated* state or *inactive* state. Each activated user u has different preference scores against the query, denoted as $\phi(u, Q)$. If user u is activated, then it has a chance with a probability $p(e)$ to activate his/her inactive neighbor v. Since the *IC* model is widely used, we also utilize the adapted *IC* influence propagation model.

2.3 Weighted Reverse Influence Set (RIS) Sampling

The *Reverse Influence Set* (RIS) [3] sampling technique is the start-of-the-art solution to the *IM* problem. By reversing the influence diffusion direction and conducting reverse *Monte Carlo* sampling, *RIS* can significantly improve the theoretical run time bound. For better understanding, we first introduce the concept of *Reverse Reachable* (*RR*) set.

Definition 1 (Reverse Reachable Set). *Suppose a user v is randomly selected from V. The reverse reachable set of v is generated by first sampling a graph g from G, and then taking the set of users that can reach to v in g.*

The *RIS* technique contains two phases: (1) Generate θ random RR sets from G; (2) Select a set S of users to cover the maximum number of RR sets generated above by transforming it to the *maximum coverage* problem [28].

The weighted reverse influence sampling (WRIS) [18] is adapted from RIS to meet the *KB-TIM* query. To differentiate the sampling probability of targeted users from non-targeted users, it defines the sampling probability for v w.r.t Q as

$$p_s(v, Q) = \frac{\phi(v, Q)}{\sum_{v' \in V} \phi(v', Q)} = \frac{\phi(v, Q)}{\phi_Q}, \tag{3}$$

where $\sum_{v' \in V} \phi(v', Q)$ is denoted as ϕ_Q. Thus, the *WRIS* method contains three phases: (1) Sample θ number of users from V with a probability of $p_s(v, Q)$ for any user v; (2) For each user v sampled, generate a RR set R_v for v; and (3) Select a set S of users to cover the maximum number of RR sets generated above based on the *maximum coverage* problem.

2.4 Pre-Computed the RR Set for Every Keyword

Theorem 1. *[18] Given a query Q, let θ_w^Q be the number of RR sets sampled by a probability of $p_s(v, w)$ w.r.t each user v and a keyword w. Then θ^Q is the total number of RR sets. $\theta^Q = \sum_{w \in Q} \theta_w^Q$. If $\theta^Q \geq \theta$ and $\frac{\theta_w^Q}{\theta^Q} = p_w$, we can achieve the same theoretical bound of the traditional RIS algorithm.*

In [18], for any keyword $w \in Q$, $|V_w|$ is the number of users with keyword w, $K = min\{|V|/2, |V_w|\}$, if we choose $\theta_w = (8 + 2\epsilon)(\sum_{v \in V} tf_{w,v}) \cdot \frac{ln|V| + ln\left(|V|K\right) + ln2}{E[\sum_{v \in V_w} \phi(v,Q)] \cdot \epsilon^2}$, then we have $\theta_w \geq \theta^Q \cdot p_w$. Thus, if we pre-compute the RR sets w.r.t a keyword $w \in Q$ in an offline step and merge the relevant RR sets at query processing, then we need to build θ_w number of RR sets for each keyword $w \in Q$. R_w is the index of RR sets sampled with probability $p_s(v, w)$ for the users in V. There are θ_w number of RR sets (denoted as R_w) for each keyword w. For each user $v \in R_w$, we maintain an inverted list L_w to indicate which RR sets contain v w.r.t. w.

Example 2. Fig. 2 shows an example of the RR sets for keywords "Note8" and "car". Suppose $\theta_{Note8} = 9$ and $\theta_{car} = 6$. We need to get 9 random sampled RR sets for "Note8" and 6 for "car". Thus, L_{Note8} and L_{car} are maintained from users to RR sets. For the query "Note8", $\{e, i\}$ is the result to the *HIM* query using [18] because $L_{Note8}(e)$ and $L_{Note8}(i)$ covers all sampled RR sets in R_{Note8}.

3 Holistic Influence Maximization

In this section, we formulate the problem of *holistic influence maximization (HIM)* for targeted advertisements in a spatial social network $G = (V, E, T, L)$.

RR_1	a, e
RR_2	b, a, e
RR_3	h, e
RR_4	d, b, a, e
RR_5	i
RR_6	e
RR_7	i, c
RR_8	b, a, e, g
RR_9	d, c, b, a, e

R_{Note8}

a	RR_1, RR_2, RR_4, RR_8
b	RR_2, RR_4, RR_8
c	RR_7
d	RR_4, RR_9
e	RR_1, RR_2, RR_3, RR_4, RR_6, RR_8, RR_9
f	
g	RR_8
h	RR_3
i	RR_5, RR_7

L_{Note8}

RR_1	d, b, a
RR_2	d, b
RR_3	b
RR_4	b, g
RR_5	c, g, e
RR_6	i, c

R_{Car}

a	RR_1
b	RR_1, RR_2, RR_3, RR_4
c	RR_5, RR_6
d	RR_1, RR_2
e	RR_5
f	
g	RR_4, RR_5
h	
i	RR_6

L_{Car}

Fig. 2. Example of RR sets for keyword *Note8* and *car*.

3.1 Influence Diffusion Through Spatial Closeness

Motivated by [11], we believe that recommendation may be influenced by location-based ratings that are *spatially* close to the users. The property, called *preference locality*, means that users have a certain probability to influence their *spatially-close* users with similar interests. Now, the challenging question is how to effectively measure the *spatial-closeness* and *similar-interests* between users. One possible way is to give the reverse weight to the users based on their *spatial-distance* directly. For two pairs of users, their spatial-closeness would be the same if their spatial-distances are the same. However, this straightforward way is not practical because there may be a large number of users with similar interests to a user in a crowded region within a small radius. It is not possible that such users can equally influence each other. In reality, an ordinary person usually contacts with a limited number of persons around him/her in daily life. Besides a new spatial-closeness metric, we also need to consider the users' similarity w.r.t. the online query in the targeted advertisement. If two users have more common interests w.r.t. the query, then they may have a higher influence on each other. Otherwise, they may not influence the advertisement even if they are very close to spatially. Besides, it is not practical to measure users' interest similarity by only using the common keywords in their interests. For instance, a user has interest about "movie" while another user has interest about "song". In this case, given a query as {movie}, it is more practical to consider that they should have a common interest with a certain probability because "movie" is highly correlated with "song" for movie or song fans.

Definition 2 (Users' Interest Similarity). *Given two users u and v with their interests $T_u = \{T_u^1, T_u^2, \ldots\}$ and $T_v = \{T_v^1, T_v^2, \ldots\}$, the interest similarity of u and v is measured by*

$$Sim(u,v) = \frac{|T_u \cap T_v|}{|T_u \cup T_v|} + \frac{\sum\{C_{t_x,t_y} | t_x \in \overline{T}_u \wedge t_y \in \overline{T}_v\}}{|\overline{T}_u| * |\overline{T}_v|}, \tag{4}$$

where $|T_u \cap T_v|$ represents the number of common keywords between T_u and T_v, $|T_u \cup T_v|$ is the total number of different keywords in T_u and T_v. In addition, $\overline{T}_u = T_u \setminus (T_u \cap T_v)$, $\overline{T}_v = T_v \setminus (T_u \cap T_v)$. For the non-common keywords t_x and t_y, C_{t_x,t_y} represents their correlation probability regarding the users' interest space in G. It is measured as

$$C_{t_x,t_y} = \frac{|\{u | u \in V \wedge t_x \in T_u \wedge t_y \in T_u\}|}{|\{u | u \in V \wedge (t_x \in T_u \vee t_y \in T_u)\}|}. \tag{5}$$

Definition 3 (Query based Users' Interest Similarity). *Given two users u and v with their interests $T_u = \{T_u^1, T_u^2, \ldots\}$ and $T_v = \{T_v^1, T_v^2, \ldots\}$, the interest similarity of u and v w.r.t. a keyword query Q is measured by*

$$Sim(u,v|Q) = \frac{\sum\{C_{t,Q} | t \in T_u \cap T_v\}}{|T_u \cup T_v|} + \frac{\sum\{C_{t_x,t_y} * max\{C_{t_x,Q}, C_{t_y,Q}\} | t_x \in \overline{T}_u \wedge t_y \in \overline{T}_v\}}{|\overline{T}_u| * |\overline{T}_v|}, \tag{6}$$

where $C_{t,Q} = max\{C_{t,t'} | t' \in Q\}$ and $C_{t,t'}$ can be calculated using Equation (5).

Definition 3 adapts users' similarity w.r.t. the given keyword query for targeted advertisements. Particularly, in $C_{t,Q}$, we use the most suitable keyword in Q to measure the correlation between a keyword t and the query Q. To measure the correlation of t_x and t_y w.r.t. Q, we re-weight their correlation C_{t_x,t_y} by using the maximum likelihood $max\{C_{t_x,Q}, C_{t_y,Q}\}$ of their independent correlation to Q.

Definition 4 (Reverse m Nearest Neighbor). *For a query user u, the reverse m nearest neighbors of u are the users for which u is one of their m nearest neighbors* (NN) *in* euclidean space *[29]. In this case, it says that u has high influence on these users in* RmNN(u).

To handle the spatial closeness and interests w.r.t. a query in an effective way, we develop the spatial diffusion by extending *reverse m nearest neighbors* (Definition 4). Generally, a user u has a high influence on the users in *RmNN* (u). Therefore, we have the following definition.

Definition 5 (Spatial Diffusion Metric SimRmNN). *Given a social network $G = (V, E, T, L)$ and a keyword query Q, a seed user u can activate his inactive neighbors based on the query based users' interest similarity and the RmNN where m is a user-specified parameter to control the spatial closeness of propagating spatial influence w.r.t. Q.*

$$SimRmNN(u) = \{v | v \in V \wedge Sim(u,v|Q) \geq \tau \wedge u \in mNN_{sim}^{Q}(v)\}. \tag{7}$$

where $Sim(u,v|Q) \geq \tau$ means that the interest similarity of u and v is no less than a system user-specified threshold τ. $u \in mNN_{sim}^{Q}(v)$ means that u is one of the m nearest τ-similar neighbors of v in G.

From Definition 5, we can see that *SimRmNN* is a dynamic metric, which is different from *RmNN* that is static. *SimRmNN* provides the metric to judge whether or not a user can activate his inactive neighbors in spatial networks.

3.2 Holistic Influence Diffusion Model

In this subsection, we devise the *holistic influence diffusion* model based on the classic *IC* model [10] for analyzing the influence diffusion of users in spatial social networks. Given a seed user u in G, its influence can be propagated via social diffusion as discussed in Section 2.2, meanwhile, the influence can also be propagated via spatial diffusion as discussed in Section 3.1. Similar to *IC* model, all users are inactive initially except u. For each edge $e = (u, v)$, u has

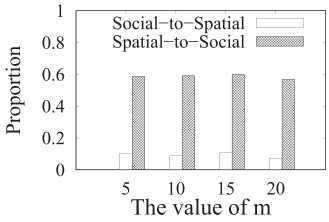


Fig. 3. The statistical comparison of influence propagation from Spatial-to-Social and Social-to-Spatial using Weibo dataset.

preference scores $\phi(v, Q)$ against the query Q and has a chance with the probability $p(e)$ to activate his inactive neighbor v. At the same time, u also has a chance with the probability $\frac{1}{SimRmNN(u)}$ and has preference scores $\phi(u, Q)$ to activate his inactive spatially-close neighbors because u has a choice with the probability $\frac{1}{SimRmNN(u)}$ to contact one of his neighbors in $SimRmNN(u)$. The users who are activated by spatial influence can further activate their social users via social influence, and activate their spatial neighbors via spatial influence. But for the other users who are activated only by social influence, they can only activate their inactive neighbors via social influence. This assumption meets with the reality that it is very likely for social users to share their offline findings on their social accounts, but the likelihood would be lower for them to discuss their online findings with their offline neighbors often. The activation procedure terminates until no inactive user can be activated.

In the holistic influence diffusion model, the social influence propagates in a "pull" style while the spatial influence propagates in a "push" style. To allow system users to control the spatial influence impact, a parameter η is used as the spatial activation threshold. v can be activated by u if $v \in SimRmNN(u)$ and $\frac{1}{SimRmNN(u)} \geq \eta$.

To analyze the influence diffusion of whether the influence of the spatially activated users can be propagated to online, and whether the influence of the socially activated users can further diffuse to offline, we have made a statistic experiment using *weibo4* dataset.[1] Fig. 3 shows that the probability of propagating information from spatially activated users to their online friends is significantly higher than the probability of propagating information from socially activated users to their offline neighbors. This verified our assumption that spatially activated users will continue to propagation their information to their social friends, but not vice versa.

3.3 Problem Definition

Based on the discussion and definitions above, we formally define our *holistic influence maximization* for targeted advertisements in spatial social networks. The goal of *HIM* is to find the minimum seed user set S whose influence can completely cover all the target users in V w.r.t. a keyword query Q expressed for a targeted advertisement.

Definition 6 (Holistic Influence Maximization). *Given a set S of seed users, a keyword query Q, and $I_h(S)$ is the holistic influenced user set of S, the expected holistic influence spread of S in G can be computed as*

$$E[I_h^Q(S)] = E\left[\sum_{v \in I_h(S)} \phi(v, Q)\right] = \sum_{v \in V}\{1 - \prod(1 - p(S \mapsto v)) \cdot (1 - p'(S \mapsto v))\} \cdot \phi(v, Q), \quad (8)$$

$p(S \mapsto v) = 1 - \prod_{u \in S}\{1 - p(u \mapsto v)\}$ represents the social influence of S to v. $p'(S \mapsto v) = 1 - \prod_{u \in S}\{1 - p'(u \mapsto v)\}$ represents the spatial influence of S to v. $p(u \mapsto v) = \prod p(e_i)$ that is calculated by multiplying the probabilities on the edges of the most influential path from u to v. $p'(u \mapsto v) = \prod \frac{1}{SimRmNN(u_i)}$ that is calculated by multiplying the spatial diffusion probabilities on the nodes along the SimRmNN *sequence from u to v.*

Based on Definition 1 in Section 2.3, if a user u appears in an RR set generated for a user v, then u can reach v via a certain path in G. As such, u should have a chance to activate v if we run an influence propagation process on G using u as a seed user. Thus, if a user u appears in a large number of RR sets, then it would have a high probability to activate more users. In this case, u's expected influence spread should be large. Borgs *et al.* [3] have proved that the probability of selecting influential seed users using RR sets is equivalent to using the sampled instance graphs. In addition, Ohsaka *et al.* [23] further described that the number of users influenced by a seed set S, is equal to the proportion of the number of S's covered RR sets multiplied by the total users in the social network. In other words, define $C(S)$ as the number of covered RR sets by seed users set S, $|I(S)|$ is the number of influenced users and θ is the total number of sampled RR sets, then $|I(S)|$ can be approximated by $n \cdot \frac{C(S)}{\theta}$. Later on, Li *et al.* [18] proposed the weighted RIS sampling method to generate the RR sets to solve the targeted IM problem. Moreover, under the generated RR sets, they also verified that the number of influenced target users has at least $1 - n^{-1}$ probability to equal with the proportion of covered RR sets multiplied by the total target users. If the seed users set can cover all RR sets, then all the targeted users will be activated by the seed users set with the theoretical guarantee. Therefore, we have the following problem definition.

Problem Statement. Given a social network G and a keyword query Q, the *HIM* query problem aims to find a minimum seed user set S from V, while the holistic influenced user set of S, $I_h(S)$, can cover all target user $u \in V$ with $\phi(u, Q) \neq 0$.

Given t sets $S_1, \ldots, S_t$ where $\mathcal{S} = \{S_1 \cup \ldots \cup S_t\}$ and $|\mathcal{S}| = n$, the set covering problem aims to select a minimum number of sets to cover all the n objects in $\mathcal{S}$. Similarly, given a social network G and a constant k, the *IM* problem under the *IC* model asks for a size-k seed set S with the maximum expected spread $E[I(S)]$, which is well-known as the *maximum covering* problem. However, our *HIM* query problem aims to find the minimum size seed set to activate all target users, which belongs to the *set covering* problem rather than *maximum covering* problem. Therefore, the *HIM* problem and the *IM* problem are different in problem settings.

Property 1. *The* HIM *query problem is a* submodular *and* non-decreasing *set covering problem.*

1. http://www.cnpameng.com/

Proof. Given a social network G and a keyword query Q, for each user u with $\phi(u,Q) \neq 0$, $I_h(u)$ is denoted as the influenced user set of u in the *HIM* query problem. In other words, there is a list of influenced user sets $\{I_h(u)\}$ for each candidate seed user $u \in V$. Therefore, the *HIM* query problem can be transformed to select the minimum number of sets from $\{I_h(.)\}$, and their union can cover all target users in V_Q, which is a typical set covering problem. Moreover, $I_h(u)$ represents the holistic influenced user set of seed user u, and $I'_u(S) = I_h(S) \cap I_h(u)$ denoting the common influenced users that are influenced by S and u independently. For two seed user sets $S \subseteq T$, we have $I_h(S) \subseteq I_h(T)$, then $I'_u(S) = I_h(S) \cap I_h(u) \leq I_h(T) \cap I_h(u) = I'_u(T)$. Therefore, we have $I_h(S \cup v) - I_h(S) = I_h(u) - I'_u(S) \geq I_h(u) - I'_u(T) = I_h(T \cup u) - I_h(T)$. Thus, for any additional seed user candidate $u \in V \setminus T$, we have $I_h(S \cup u) - I_h(S) \geq I_h(T \cup u) - I_h(T)$ and $I_h(S \cup u) \geq I_h(S)$. $I_h(.)$ is proved to be *submodular* and *nondecreasing*. So *HIM* query problem is a *submodular* and *nondecreasing* set covering problem. □

The set covering problem has been proved as a classic NP-hard problem. To solve the problem, Wolsey [28] analyzed it and proposed the greedy algorithm with a theoretical approximation ratio. Based on [28], we induce a greedy method analysis for solving the *HIM* problem.

Property 2. *The* HIM *query problem can be solved by greedy algorithm with* $(\min\{t_1, t_2, t_3\})$*-approximation ratio, where* $t_1 = 1 + \ln \frac{|V_Q|}{|V_Q| - |I_h(S_{r-1})|}$, $t_2 = 1 + \ln \frac{|I_h(S_1)|}{|I_h(S_r)| - |I_h(S_{r-1})|}$, $t_3 = 1 + \ln \max\{\frac{|I_h(u)|}{|I_h(S_\ell \cup \{u\})| - |I_h(S_\ell)|}\}$ *with* $1 \leq \ell \leq r$, $u \in V_Q$, $|I_h(S_\ell \cup u)| - |I_h(S_\ell)| > 0$, *$S_\ell$ is the seed set found by greedy algorithm at the end of iteration ℓ ($1 \leq \ell \leq r$).*

Proof. Based on Property 1, the *HIM* query problem is a *submodular* and *nondecreasing* set covering problem. From [20], [28], suppose $f(.)$ is a *nondecreasing* and *submodular* function defined on subsets of a object set U, a greedy algorithm can be used to solve the set covering problem with $(\min\{t_1, t_2, t_3\})$-approximation ratio, where $t_1 = 1 + \ln \frac{f(U) - f(\emptyset)}{f(U) - f(S_{r-1})}$, $t_2 = 1 + \ln \frac{f(S_1) - f(\emptyset)}{f(S_r) - f(S_{r-1})}$, $t_3 = \ln \max\{\frac{f(x) - f(\emptyset)}{f(S_\ell \cup x) - f(S_\ell)} | 1 \leq \ell \leq r, x \in U, f(S_\ell \cup x) - f(S_\ell) > 0\}$, and the greedy algorithm terminates after r iterations while S_ℓ denote the seed set at iteration $\ell \in [1, r]$ ($S_0 = \emptyset$).

In *HIM* query problem, we denote S_i as the seed set at the end of iteration i in the greedy algorithm where $i = 1, 2, .., r$, $I_h(.)$ is a *submodular* function, and $I_h(S_r) = V_Q$. According to the theoretical analysis of greedy algorithm for the set covering problem in above, we can derive the conclusion that our proposed greedy algorithm of *HIM* query problem has a $(\min\{t_1, t_2, t_3\})$-approximation, where $t_1 = 1 + \ln \frac{|V_Q|}{|V_Q| - |I_h(S_{r-1})|}$, $t_2 = 1 + \ln \frac{|I_h(S_1)|}{|I_h(S_r)| - |I_h(S_{r-1})|}$, and $t_3 = 1 + \ln \max\{\frac{|I_h(u)|}{|I_h(S_\ell \cup \{u\})| - |I_h(S_\ell)|} | 1 \leq \ell \leq r, u \in V_Q, |I_h(S_\ell \cup u)| - |I_h(S_\ell)| > 0\}$. □

Although the greedy algorithm can solve the *HIM* query problem with the theoretical guarantee, the estimation of $E[I_h(u)]$ for each user u in G may incur significant computation overheads if we directly compute the influenced user set for each seed candidate. Like the other *IM* works [5], [10], we also use the RR sets to solve the *HIM* query problem.

Property 3. *The* HIM *query problem can be answered by selecting the minimum seed set $S \subseteq V_Q$ and its counterpart S' being able to cover all the RR Sets $\{R_w\}$ with $w \in Q$ where the social influence seed user set and the corresponding spatial influence seed user set are denoted as S and S' respectively.*

Proof. Given a seed user set S, based on the *HIM* model, we initially compute its spatially activated user set S'. Then, we can identify the influenced user set by $\{S \cup S'\}$, i.e., the users can be activated by either social influence seed set S or spatial influence seed set S', or both of them. Thus, we have $E[I_h^Q(S)]$ equals $E[I^Q(S \cup S')]$. Besides, from the weighted reverse influence sampling, let $F_{\theta^Q}(u)$ denote the number of sampled RR sets covered by a user u, then $E[F_{\theta^Q}(u)]$ equals the probability that u intersects a random RR set from R_w with keyword $w \in Q$, while $\frac{E[I^Q(u)]}{|V_Q|}$ equals the probability that a randomly selected target user in an influence propagation process on G. Based on [26], we have $|V_Q| \cdot \frac{F_{\theta^Q}(S \cup S')}{\theta^Q}$ is an accurate estimator of any user set S's expected spread under *HIM* model with probability $1 - |V|^{-1}$, when θ^Q is sufficiently large. Therefore, *HIM* model can be answered by finding the minimum seed set S and its counterpart S' covering all RR sets $\{R_w\}$ with $w \in Q$. Based on the above discussion, it can be seen that the *HIM* with the target RR sets based covering seed set selection is also a *set covering* problem. In addition, the function $F_{\theta^Q}(\cdot)$ satisfies the *submodular* and *nondecreasing* properties based on Property 2. Therefore, the RR sets based greedy algorithm of solving the *HIM* query problem can be proved to have the $(\min\{t_1, t_2, t_3, t_4\})$-approximate ratio, where $t_1 = 1 + \ln \frac{\theta^Q}{\theta^Q - |F_{\theta^Q}(S_{r-1} \cup S'_{r-1})|}$, $t_2 = 1 + \ln \frac{|F_{\theta^Q}(S_1 \cup S'_1)|}{|F_{\theta^Q}(S_r \cup S'_r)| - |F_{\theta^Q}(S_{r-1} \cup S'_{r-1})|}$, $t_3 = 1 + \ln \max\{\frac{|F_{\theta^Q}(u \cup u')|}{|F_{\theta^Q}(S_\ell \cup S'_\ell \cup u \cup u')| - |F_{\theta^Q}(S_\ell \cup S'_\ell)|} | 1 \leq \ell \leq r, u \in V_Q, |F_{\theta^Q}(S_\ell \cup S'_\ell \cup u \cup u')| - |F_{\theta^Q}(S_\ell \cup S'_\ell)| > 0\}$, $t_4 = \mathcal{H}(\max_{u \in V_Q}\{F_{\theta^Q}(u \cup u')\})$ where $\mathcal{H}(d) = \sum_{i=1}^{d} \frac{1}{i}$, and S_i is the seed user set selected by greedy algorithm at the end of iteration $i (1 \leq i \leq r)$, S'_i (or u') is the spatial activated user set of S_i (or u). □

4 Our Solutions to *HIM* Query Problem

In this section, we first propose a baseline solution to solve the *HIM* query problem. Next we analyze its computational bottleneck and develop an improved solution. Finally, we devise a novel spatial-social index R_{SS} and R_{SS} based algorithm to efficiently solve the *HIM* query problem.

4.1 Baseline Solution

The baseline solution is a greedy algorithm that finds the minimum number of seed users one by one based on the users' holistic diffusion spread. The main idea is to first load the precomputed RR sets relating to the given keyword query Q. It then computes the interest similarity for every pair of users w.r.t Q in G and maintain the information for computing the spatial influence of users. After that, it calculates the number of spatially activated users for each user in V and checks if they can be successfully activated based on the threshold η. If a user is

activated via spatial influence, then it needs to further exploit the newly activated user's neighbors. After each user has been computed, the first seed user can be selected based on the current holistic influence spread (i.e., RR sets) of users and its corresponding RR sets is removed. The above procedure is repeated until all the related RR sets are removed. Algorithm 1 presents the pseudo code.

Algorithm 1. Baseline Algorithm

Input: A social network $G = (V, E, T, L)$, a keyword query Q, an interest similarity threshold τ, a spatial activation threshold η
Output: The minimum seed user set S_Q
1 **for** $w \in Q$ **do**
2 R_w^Q={RR_i:list of users} obtained $\theta^Q \cdot p_w$ number of RR sets from R_w RR sets
3 Generate the inverted user-to-RR list L_w^Q from R_w^Q
4 **for** *any pair of users* $u, v \in V$ **do**
5 **if** $Sim(u, v|Q) \geq \tau$ *and* $u \in mNN_{sim}^Q(v)$ **then**
6 $SimRmNN(u) += \{v\}$
7 **while** $R_w^Q \neq \phi$ **do**
8 Get the sorted list L of users $\in V$ in descendent order of their RR set number $|\cup \{L_w^Q | w \in Q\}|$
9 **for** *each user* $u \in V$ **do**
10 Initialize temporary set $S^u = \{u\}$
11 **for** *each user* $u' \in SimRmNN(u)$ **do**
12 Add u' into S^u
13 **for** *each user* $v \in S^u$ **do**
14 Visit and mark v as visited in S^u
15 **for** *each user* $v' \in SimRmNN(v)$ **do**
16 **if** $\frac{1}{|SimRmNN(v)|} \cdot \phi(v', Q) \geq \eta$ **then**
17 Add v' into S^u
18 Update u's position in L using the number of RR sets covered by $\{u \cup S^u\}$
19 Get and add the first user from L into S_Q
20 Remove the RR sets of $\{u \cup S^u\}$ from R_w^Q
21 **return** S_Q

The time complexity of Algorithm 1 is calculated as follows. In Lines 1-3, it needs to scan and load R_w^Q for $w \in Q$ by taking $O(|R_w^Q| \cdot |Q|)$. In Lines 4-6, the for-loop operation takes $O(|V|^2)$, computations $Sim(u, v|Q)$ takes $O(|T_u| \cdot |T_v| \cdot |V|)$, and mNN search takes $O(|V|)$. Thus, the time complexity of this part is $O(|V|^2 \cdot (|T_u| \cdot |T_v| \cdot |V| + |V|))$. In Lines 9-18, the computational complexity is bounded by $O(|V| \cdot (|V| + |V| \cdot log|V|))$. Thus, the time complexity of Lines 7-20 is $O(|R_w^Q| \cdot (1 + log|V|) \cdot |V|^2)$. By aggregating the above procedures, the total time complexity of Algorithm 1 is $O(|R_w^Q| \cdot |Q| + (|T_u|^2 + 1) \cdot |V|^3 + (1 + log|V|) \cdot |R_w^Q| \cdot |V|^2)$.

Based on the time complexity above, the most challenging computational part in Algorithm 1 is to check the interest similarity for any pair of users in V w.r.t. Q. Since the spatial influence diffusion depends on the interest similarity and the *RmNN* relations together, only a few nearby similar neighbors for a given user may require to be visited. Therefore, we develop an improved solution to exploit the spatial influence spread for a user using local search strategy.

4.2 Improved Solution

In this subsection, we propose an improved solution to reduce the most unnecessary computation of the interest similarity between users, during the process of computing $SimRmNN(u)$. For each user $v \in V$, we first compute the distance $D(u, v)$ between users u and v, then we check whether or not there exists m number of users $v' \in V$ satisfying two conditions (1) $D(v, v') < D(u, v)$; (2) $Sim(v, v'|Q) \geq \tau$. If v satisfy with the above conditions, then v could not be the member of $SimRmNN(u)$. Otherwise, $v \in SimRmNN(u)$.

Algorithm 2. Improved Algorithm

1 Lines 1–3 in Algorithm 1
2 Load the R-$tree$ index R
3 **for** *each leaf node* $R_i \in R$ **do**
4 **for** *each users* $u \in V$ **do**
5 Initialize $R_i(u)$ as *unsatisfied*
6 **while** $R_w^Q \neq \phi$ **do**
7 Get the sorted list L of users $\in V$
8 **for** *each user* $u \in V$ **do**
9 Initialize $S^u = \{u\}$
10 **while** $u \in S^u$ *and* u *is not visited* **do**
11 Visit u and set it as visited in S^u
12 Call *Compute_SimRmNN*(u) to compute $SimRmNN(u)$
13 **for** *each user* $u' \in SimRmNN(u)$ **do**
14 Add u' into S^u
15 **for** *each user* $v \in S^u$ **do**
16 Visit and mark v as visited in S^u
17 **for** *each user* $v' \in SimRmNN(v)$ **do**
18 **if** $\frac{1}{|SimRmNN(v)|} \cdot \phi(v', Q) \geq \eta$ **then**
19 Add v' into S^u
20 Update u's position in L using the number of RR sets covered by $\{u \cup S^u\}$
21 Lines 19-20 in Algorithm 1
22 **return** S_Q

Algorithm 2 presents the details of our *improved solution*. Compared with Algorithms 1, 2 utilizes a different way to compute the query based on users' interest similarity. At Lines 9-19, it first computes the $SimRmNN(u)$ for a given user u by calling Function $Compute\_SimRmNN(u)$. After all users in $SimRmNN(u)$ are found, it further considers the spatial activation process. Once there exist many inactive users that can be successfully activated by the activated users under the spatial diffusion model, then it exploits these newly activated users in Lines 15-19. The algorithm terminates until the selected seed users cover all RR sets.

Algorithm 3 presents the procedure of computing the $SimRmNN(u)$ of a user u. The main idea of Algorithm 3 is to judge whether the user $v \in \{V \setminus u\}$ is the member of $SimRmNN(u)$. For each user $v \in \{V \setminus u\}$, if existed m similarity neighbors v' with distance $D(v, v') < D(u, v)$, then v is not the member of $SimRmNN(u)$ and vice versa. To do this, we first build a R-$tree$ index and define some bounds to prune the search space. Then, we set the $D_{min}(v, R_j)$ represents the minimum possible distance between v and any users in the subtree of R-$tree$ node R_j, and $D_{max}(v, R_j)$ denotes the maximum possible distance between user v and any users in the subtree of R_j. For each user $v \in \{V \setminus u\}$, we maintain a min-heap P_{queue} with the R-$tree$ node visited so far, sorted by their D_{max} with user v. Next we pop out the first node R_i from P_{queue} with minimum value of $D_{max}()$. If the pop node R_i is not leaf node, we insert the child node of

R_i into P_{queue}. Otherwise, we first judge whether $D_{max}(v, R_i) < D(u, v)$. If so, we further accumulate the number of similarity users between v and $v' \in R_i$, and then use *count* to represent it. Finally we decide whether *count* is satisfied with no less than m. We set the label of $R_i(v)$ to *satisfied* while $count \geq m$. If $count < m$, then we continue to exploit the user, whose distance to v is shorter than $D(u, v)$. The above procedures are shown in Lines 7-30. At Lines 5-6, when we compute the $SimRmNN$ w.r.t. u, for the user v, if the label $R_j(v)$ of v is *satisfied* and $D_{max}(v, R_j) < D(u, v)$, then v is directly pruned from the candidate of $SimRmNN(u)$ and do not process any users' interest similarity computation. Compared with Algorithm 1, the running time cost of the Algorithm 2 is clearly decreased, because of the considerably number of reduced user's interest similarity computation.

Algorithm 3. $Compute\_SimRmNN(u)$

```
1  for each user v ∈ {V \ u} do
2    count ← 0
3    Initialize a min-heap P_queue accepting entries of the R-tree
     node R, P_queue is sorted by D_max(v, R)
4    Insert the root of R-tree into P_queue
5    if existed R_j ∈ R where R_j(v) is satisfied and D_max(v, R_j) <
     D(u, v) then
6      count ← m
7    while P_queue is not empty do
8      Pop the first node R_i from P_queue
9      if count ≥ m then
10       break
11     else
12       if R_i is leaf node then
13         if D_max(v, R_i) < D(u, v) then
14           for each user v' in R_i do
15             if count ≥ m then
16               set R_i(v) as satisfied
17               break
18             else if Sim(v, v'|Q) ≥ τ then
19               count ← count + 1
20         else
21           if D_min(v, R_i) < D(u, v) then
22             for each user v' in R_i do
23               if count ≥ m then
24                 break
25               else if D(v, v') < D(u, v) and Sim(v, v'|Q) ≥ τ
                 then
26                 count ← count + 1
27       else
28         Insert the child node of R_i into P_queue
29   if count < m then
30     SimRmNN(u)+ = v
```

4.3 Proposed Spatial Social Index R_{SS}

Although the *improved algorithm* can reduce computational cost compared to the *baseline algorithm*, it is still required for the *improved algorithm* to run the *SimRmNN* query many times. In the worst case, it needs to run for every user in V, which results in the time complexity of computing *SimRmNN* as $O(|V|^2)$. Therefore, it is desirable to design a novel index to reduce computational cost further. Besides, for answering the *HIM* query, we also need a novel index to maintain the pre-computed RR sets, the location information of users, and users' interests. So we develop a spatial social index based on *R-tree*, denoted as R_{SS} index. First, we initialize R_{SS} index by inserting all the users in V into a *R-tree* index based on users' location information. Next, for each user in a leaf node, an inverted file is created for indexing each keyword w (v user's interests) and the related RR sets for every user v that is contained in this leaf node. We then generate the inverted file L_w from the number of RR sets w.r.t. w and v where v is the entry of L_w. Thus, each user consists of three parts (w, $|L_w|$, L_w). Each internal node R_i contains each keyword and the maximal number of RR sets of individual users w.r.t. the keyword $R_i(L_w)$. In addition, for each leaf node and internal node R_i, we also maintain one additional information $D_{max}(R_i, R_j)$ where R_j is any leaf node $\notin R_i$ and it has the smallest maximum distance to R_i. R_{SS} index is built based on pre-computed RR sets and keywords using the preliminary in Section 2.

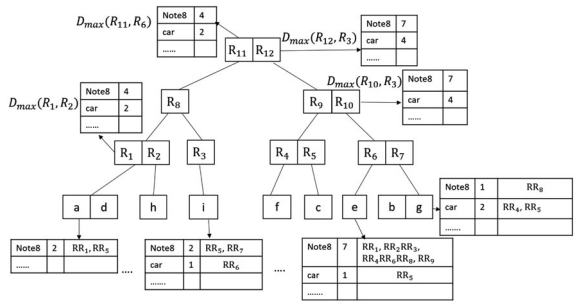


Fig. 4. Example of R_{SS} index structure.

Example 3. Fig. 4 presents the example of R_{SS} index structure. For each user, e.g., a, R_{SS} maintains the keywords as their interests, e.g., *Note8*, and their lists of RR sets, e.g., L_{Note8}. For the leaf nodes and internal nodes, R_{SS} also maintains additional information of $D_{max}()$. For instance, R_2 is a leaf node and has the smallest maximum distance to R_1. So we have $D_{max}(R_1, R_2)$ for R_1. For an internal node R_{11}, the leaf node with the smallest maximum distance to R_{11} is R_6. So $D_{max}(R_{11}, R_6)$ is maintained for R_{11}.

Property 4 (SimRmNN Pruning Rule). *Given a user $u \in R_x$, the users in R_i can not be in* SimRmNN(u) *if $D_{min}(u, R_i) > D_{max}(R_i, R_j)$ and $|\{v | v \in R_j \wedge Sim(u, v|Q) \geq \tau\}| \geq m$.*

Proof. $D_{min}(u, R_i)$ is the minimum distance from u to any user $u' \in R_i$. Since $D_{min}(u, R_i) > D_{max}(R_i, R_j)$, we must have that the distance between $v \in R_j$ and $u' \in R_i$ is smaller than the distance between u and u'. Thus, there are at least $|R_j|$ users who are closer to u' than u. And R_j contains at least m users similar to u. It says that u can not become one of m nearest similar neighbors of $u' \in R_i$, i.e., R_i does not contain any user in $SimRmNN(u)$. So R_i can be pruned when we compute *SimRmNN* for u. □

Property 5 (Upper Bound of RmNN ***Size)***. *For any query point u, $RmNN(u)$ has at most $6 \cdot m$ data points in a 2-dimensional euclidean space in [25].*

4.4 Index-Based Solution

The key idea of our *index-based* solution is to find the best seed user that has the maximum holistic influence spread

w.r.t. the RR sets. It then reduces the expensive cost of computing *SimRmNN* for a large number of users by using the upper bound of *SimRmNN*. Besides, we can easily find the user with the maximum RR sets using R_{SS}. By multiplying the two upper bound values, we can work out the upper bound of holistic influence spread at this moment, by which we can safely judge whether or not a user can become the best seed user. The seed users can be selected one by one until the required number of RR sets can be achieved.

Algorithm 4. Index-Based Algorithm

Input: $G = (V, E, T, L)$, Q, τ and η
Output: The minimum seed user set S_Q
1 Load the spatial social index R_{SS}
2 Initialize $t^Q_{total} = 0$
3 **while** $t^Q_{total} < \sum_{w \in Q} \theta^Q \cdot p_w$ **do**
4 Get the user v with $max\{\sum_{w \in Q} |L_w(v)|\}$
5 Compute v's spatial activation set S^v using Line 9– 19 in Algorithm 2
6 Compute the RR sets covered by $\{v \cup S^v\}$, and record v and the set size as $(v : RR^v_{max})$
7 **for** *each user* $u \in V$ **do**
8 $RR^u_{upper} = SimRmNN\_Upper(u, RR^v_{max})$
9 **if** $RR^u_{upper} > RR^v_{max}$ **then**
10 Compute the RR sets covered by $\{u \cup S^u\}$, and record u and the set size as $(u : RR^u_{max})$
11 **if** $RR^u_{max} > RR^v_{max}$ **then**
12 $RR^v_{max} \leftarrow RR^u_{max}$ and $v \leftarrow u$
13 $t^Q_{total} += RR^v_{max}$
14 Mark the RR sets of $\{v \cup S^v\}$ as visited in R_{SS} index

Algorithm 5. SimRmNN_Upper(u, RR^v_{max})

1 Initialize P_{queue} with the maximum non-overlapped nodes R_i of R_{SS} where $u \notin R_i$, P_{queue} is sorted by $\sum_{w \in Q}\{w.count | w \in R_i\}$
2 Initialize an empty set P_u
3 **while** P_{queue} *is not empty* **do**
4 Pop the first node R_i from P_{queue}
5 **if** $6m \cdot \sum_{w \in Q}\{w.count | w \in R_i\} < RR^v_{max}$ **then**
6 Break
7 **else**
8 Get R_j using $D_{max}(R_i, R_j)$ in R_{SS}
9 Compute $D_{min}(u, R_i)$ using R_i's coordinates
10 **if** $D_{min}(u, R_i) > D_{max}(R_i, R_j)$ *And* $|\{v | v \in R_j \wedge Sim(u, v|Q) \geq \tau\}| \geq m$ **then**
11 *** R_i can be pruned based on Property 4
12 **else**
13 **if** R_i *is not leaf node* **then**
14 Add R_i's child nodes into P_{queue}
15 **else**
16 Add $\{v | v \in R_i \wedge Sim(u, v|Q) \geq \tau\}$ into P_u
17 **for** $v \in P_u$ **do**
18 Call for $SimRmNN\_Upper(v, RR^v_{max})$ with probability $\frac{1}{|P_u|}$
19 **return** $6m \cdot max\{|\sum_{w \in Q} |L_w(v)|| v \in P_u|\}$

Algorithm 4 presents a detailed procedure. For each keyword in Q, we need to cover a certain number of its corresponding RR sets based on our problem definition and preliminaries. To do this, we select the user v with the maximum social influence spread, i.e., covering the maximum number of RR sets by itself. And then we compute its spatial activation set S_v using Line 9-19 in Algorithm 2. Thus, we can have the holistic influence spread of $\{v \cup S^v\}$ by accessing their RR sets in R_{SS} index. After that, we probe the next user by calling Function *SimRmNN_Upper*(), by which we can generate the upper bound of the holistic influence spread of the best user if exists. The algorithm terminates when the selected seed users cover the required number of RR sets. These selected seed users are the minimum seed user set to the *HIM* query. The index-based algorithm has the same performance in the worst case as the improved algorithm. But in most cases, it can prune more users without computing their *SimRmNN* using the upper bound values.

Algorithm 5 presents the procedure of computing the upper bound of RR sets w.r.t. a user u. The main idea of Algorithm 5 is to prune the R_{SS} tree nodes that can not contain any potential *SimRmNN* of u using Property 4. To do this, it first finds all the tree nodes that are not overlapped and do not contain u in themselves or their leaf nodes, denoted as the *maximum non-overlapped nodes*. Following this, we pop out the first node R_i from the priority queue P_{queue}. If $6m \cdot \sum_{w \in Q}\{w.count | w \in R_i\} < RR^v_{max}$ holds, based on Property 5, then no user in R_i can reach more RR sets than v relating to RR^v_{max}. R_i and the rest of the nodes in P_{queue} can also be deleted as P_{queue} is ordered. Otherwise, it checks if R_i can be pruned by probing another node, e.g., R_j that has the smallest maximum distance to R_i, i.e., $D_{max}(R_i, R_j)$ maintained in R_i in R_{SS} index. The procedures are shown in Lines 8-16. For the rest of the users who cannot be filtered out, they are added as the potential *SimRmNN* set P_u. Each user in this set has a chance to be activated with the probability $1/P_u$, which may lead to adding more potential users. After finding all the potential *SimRmNN*, we identify the maximum number of RR sets for each of these users as the social influence upper bound of a potential seed user, i.e., $max\{|\sum_{w \in Q} |L_w(v)|| v \in P_u|\}$. Based on Property 5, we get the final holistic influence spread upper bound using Line 19.

4.5 Dynamic Maintenance of R_{SS} Index

In this subsection, we describe how to maintain our R_{SS} index efficiently with the evolution of the dynamic social network $G = (V, E, T, L)$. We consider three most common dynamics of G: edge additions, edge deletions and vertex additions. First, we describe how we update the precomputed RR sets index R_w and the inverted list L_w for every keyword w. Next, we present how we maintain the updates of the R_{SS} index when G changes.

4.5.1 Dynamic Update of R_w and L_w

Ohsaka *et al.* [23] proposed the first real-time fully-dynamic index data structure RR_w to maintain the RR sets for each keyword $w \in Q$. The index $RR_w = \{R_w, E_w\}$ is an extension of R_w index. Similar to the RR sets generation process in Section 2.3, the RR_w index is constructed by using the following reverse-BFS-like method which contains two phases: (1) Sample θ_w number of users from V with a probability of $p_s(v, Q)$ for any user v; (2) For the nth sampled user v with $n \in [1, \theta_w]$, we use the reverse-BFS method to generate a RR set $R_w(n)$ and $E_w(n)$ for v to record the visited users and visited edges respectively. Specifically, the sampled user v is the first member of $R_w(n)$ and we represent v as $R^1_w(n)$; (3)

The $RR_w = \{R_w, E_w\}$ index is constructed by $R_w = \{R_w(1), .., R_w(\theta_w)\}$ and $E_w = \{E_w(1), .., E_w(\theta_w)\}$.

We resort to the method of [23] to maintain the RR_w index while G is dynamically changed. Details are presented below, and the pseudocode is in Algorithm 6.

Edge Addition. Once we have decided to add an edge uv to G, we first add the edge uv to the current graph edge set E of G. Then, for every keyword $w \in Q$, we perform a reverse-BFS search from u. And for every RR set $RR_w(n) \in L_w(v)$, we add the visited user to $R_w(n)$ and add the visited edge to $E_w(v)$. After that, the RR_w index have updated. Finally, we update the L_w based on the updated RR sets R_w.

Edge Deletion. To delete an edge uv from G, we first delete the edge uv from E. Then, for every keyword $w \in Q$, we find the index $RR_w(n) = \{R_w(n), E_w(n)\}$ that contains edge uv, such that $uv \in E_w(n)$, and remove the user in $R_w(n)$ from which we can no longer reach $R_w^1(n)$ after deleting the edge uv. This is done by recomputing the set of vertex that can reach $R_w^1(n)$ by conducting a reverse BFS from $R_w^1(n)$. Next we delete the edge in $E_w(n)$ which map with the removed user in $R_w(n)$. Finally, we update the L_w based on the updated RR sets R_w.

Algorithm 6. The Dynamic Update of RR Sets

Input: $G = (V, E, T, L)$, Q, τ, η, new added user u, new added/deleted edge uv, pre-computed $RR_w = \{R_w, E_w\}$ index and the inverted list L_w
Output: The updated R_w and L_w

Procedure *Edge addition*
Add edge uv to edge set E of G
Generate the RR set $R_w(0)$ and $E_w(0)$ from u
for *each* i *with* $\{i | R_w(i) \in L_w(u) \wedge i \in [1, \theta_w]\}$ **do**
  Add vertex v' with $\{v' | v' \in R_w(0) \wedge v' \notin R_w(i)\}$ into $R_w(i)$
  Add edge uv' with $\{uv' | uv' \in E_w(0) \wedge v' \notin E_w(i)\}$ and uv into $E_w(i)$
Update L_w based on the updated R_w
Procedure *Edge deletion*
Delete edge uv from edge set E of G
for *each* i *with* $\{i | uv \in E_w(i) \wedge i \in [1, \theta_w]\}$ **do**
  Delete uv from $E_w(i)$
  Preform a reverse-BFS from $R_w^1(i)$ and use $R_w(0)$ to record the visited users
  delete vertex v' with $\{v' | v' \notin R_w(0)\}$ from $R_w(i)$ and delete the edge where contains v' from $E_w(i)$
Update L_w based on the updated R_w
Procedure *Vertex addition*
for $i \in [1, \theta_w]$ **do**
  return $j = i$ with minimum $T = W^{1/\phi(R_w^1(i), w)}$ where $W = random(0, 1)$
if $W^{1/\phi(u,w)} > T$ **then**
  Replace $R_w(j)$ and $E_w(j)$ by performing a reverse BFS method from u.
  Update L_w based on the updated R_w
else
  Continue
return R_w *and* L_w

Vertex Addition. To add a user u to G, we need to update the sampling target user in the RR_w index to preserve the property that each user v in G is chosen with a probability of $P_s(v, w)$ at random as a target user. Note that this is different from Ohsaka *et al.* [23] who do not consider the users' preference. The RR_w index maintenance with user addition in [23] only need to preserve the sampling target user that is chosen uniformly at random. In our work, due to the consideration of the user's preference, we additionally maintain the weighted random sampling of target users in a RR_w index when adding a user. Efraimidis *et al.* [7] proposed a weighted random sampling method that point out the direction to solve our problem. Details are given below.

Let V and $V' = \{V \cup u\}$ denote the user set of G before and after we add a new user u respectively. As mentioned in Section 2.3, we construct RR_w index from reverse-BFS operation after sampling a target user v. Obviously, for each time we chosen a target user, the probability that the target user is chosen from V is $\frac{\sum_{u \in V} \phi(u,w)}{\sum_{u' \in V'} \phi(u',w)}$, and the probability that the chosen target user is u is $\frac{\phi(u,w)}{\sum_{u' \in V'} \phi(u',w)}$. In order to ensure that this property holds, we first compute the value of $T_{v'} = W_{v'}^{1/\phi(vi,w)}$ where $W_{v'} = random(0, 1)$ for every selected target user v' in RR_w index. Then we use the threshold T to record the smallest value of $T_{v'}$ for every selected target user v' in RR_w index. For added user u, if $W_u^{1/\phi(u,w)} > T$, we use u to replace the target user with the smallest $T_{v'}$ in RR_w index and update the R_w and E_w index through conducting a reverse-BFS from u. Finally, we update L_w based on the updated RR sets R_w.

4.5.2 Dynamic Update of R_{SS} Index

After updating the RR sets R_w and the inverted list L_w, we can easily maintain our R_{SS} index when the user or the relationship between users are dynamically changed. In other words, when a vertex is added or an edge is added/deleted, we would efficiently update the attached information $(w, |L_w|, L_w)$ for every user in leaf nodes of R_{SS} index by using the method in Section 4.5.1. After that, we start to maintain our R_{SS} index.

Change of Edges. As the edge of the graph changes, we use Post-Order Traversal-like method to traverse the R_{SS} index and start from the influenced users. For each internal node R_i, once existed a traversed entry R_j of R_i where $R_j(|L_w|) > R_i(|L_w|)$, we update the value of $R_i(|L_w|) = R_j(|L_w|)$ and continue the search. Otherwise, if $R_i(|L_w|)$ does not change after visiting all influenced nodes in R_i, the traverse of this branch is stopped.

Addition of Vertex. The R_{SS} index is based on R-*tree*. Similar to R-*tree*, once the vertex v changes, the area of the internal node R_i which contains v may change. Thus, we need to recompute the value of $D_{max}(R_i, R_j)$ for every area changed internal node R_i. Besides, we update the information of the maximal number of RR sets of individual users for nodes of R_{SS} index by using the method mentioned above in the change of edges.

5 Experiments

In this section, we conduct comprehensive experiments to verify the effectiveness and efficiency of our proposed *HIM* query and the proposed algorithms. All the algorithms are implemented in Python 2.7 and run on a Windows 10 Enterprise(Intel(R) Core(TM) i7-6700 CPU@3.40 GHz and 40 GB

TABLE 1
Selected Datasets

Statistics	facebook	weibo4	weibo10	twitter
Total number of users	4.04k	58.21k	196.44k	1,671k
Total number of edges	88.23k	428.13k	1,900.46k	17,693k
Total number of posts	42.31k	824.61k	3,574.12k	37,210k

TABLE 2
Evaluation of Constructing Offline RR Sets Indices

Dataset	Disk Size(RR)	Time	Disk Size(R_{SS})
facebook	5.95 MB	13.7s	0.97 MB
weibo4	464 MB	517.87s	21.40 MB
weibo10	1,015.93 MB	1,959.77s	113.21 MB
$weibo4_1$	437 MB	489.21s	21.27 MB
$weibo4_2$	441 MB	491.10s	21.34 MB
$weibo10_1$	974 MB	1803.1s	115.76 MB
$weibo10_2$	923 MB	1746.31s	109.31 MB
twitter	23.6 GB	6.3h	1.30 GB

RAM). Since there is no ground truth dataset to verify the spatial influence diffusion, we develop a new metric to verify the actual spatial influence diffusion based on the time-varied datasets.

Datasets. We use four datasets in the experiments, as shown in Table 1. For *twitter* dataset,[2] we label the users' locations using their frequently mentioned locations in their tweets. The *facebook* dataset is obtained by crawling Facebook API. And two additional datasets are *weibo4* and *weibo10*.[3] To verify the effectiveness of *HIM* query, we generate two variants for each weibo dataset by splitting it into two consecutive months, denoted as $weibo4_1$, $weibo4_2$, $weibo10_1$, $weibo10_2$. With the support of time-varied datasets, we can verify if a user can be influenced by his spatially close neighbors by detecting the particular keyword changes in a period of time. We extracted 200 topics from each dataset, and the user profile is represented by a term vector in the topic space.

Parameters. We consider four parameters in our experiment: an integer m, query size k_Q, similarity threshold τ and spatial activation ratio η. Besides, we follow a similar method in [18] to select test keyword queries for our datasets. To evaluate the performance of our algorithms with the increasing number of keywords in a query, we vary the number of query keywords from 1 to 4 and select 50 queries from candidate keywords space for each length. Unless otherwise mentioned, the default parameter settings $m = 10$, $k_Q = 3$, $\tau = 0.05$ and $\eta = 0.01$ are used in the following evaluation. Also, we set the *influence probability* $p(e)$ as $p(e) = \frac{1}{N_v}$, where N_v is the in-degree of v. This is because our proposed methods are independent of how $p(e)$ is set and $p(e)$ is normally set to $p(e) = \frac{1}{N_v}$ in many previous *IM* works [9], [18].

5.1 Evaluation of Constructing R_{SS} Index

Table 2 presents the space cost and the time cost when we compute the RR sets for different datasets. For instance, the size of RR sets for *weibo10* is 1,015.93 MB, and its R_{SS} index is 113.21 MB. The running time of computing RR sets takes 1,959.77 seconds in the experimental configuration.

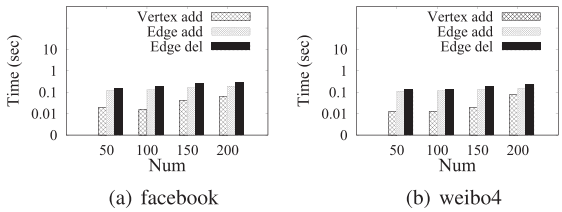


Fig. 5. Time cost of RR sets update.

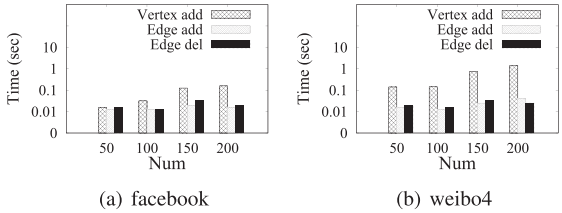


Fig. 6. Time cost of R_{SS} index update.

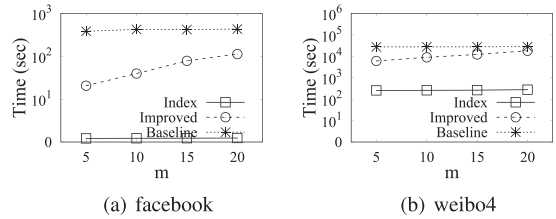


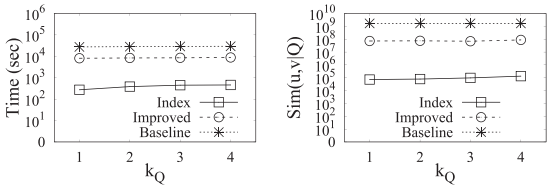
Fig. 7. Time cost of algorithms when m varies.

R_{SS} Index Maintenance. Figs. 5 and 6 present the average time cost of updating the RR sets and the R_{SS} index respectively, when the social network dynamically changed. In Fig. 5, we evaluate the running time cost of updating RR sets for three common dynamic social network changes, including vertex addition, edge addition, and edge deletion. Updating RR sets w.r.t. vertex addition takes much less time than edge addition. The deletion of edges is the most expensive operation in terms of updating RR sets. Fig. 6 shows the time cost of maintaining index R_{SS} when the social network changes. It finds that the time cost of updating the R_{SS} index increases significantly with the additions of vertices. The time cost trend is not obvious with the changes of edge addition and edge deletion. This is because only the vertex addition operation may cause the node decomposition of R_{SS}, which incurs the high cost of index maintenance.

5.2 Evaluation of Efficiency

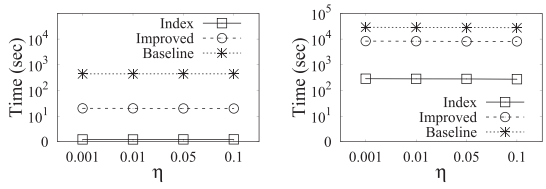
Varying m. Fig. 7 compares the running time of the approaches when we vary m. We notice that *Index* is significantly faster than *Baseline* and *Improved* in each dataset, and *Improved* performs better than *Baseline*. Besides, when we increase m, the running time of *Improved* increases significantly while *Baseline* and *Index* grow slowly. This is because only a few users need to be explored with regards to the *SimRmNN* computation in *Index* and the majority running time consumption of *Baseline* is concentrated on $Sim(u, v|Q)$ computation.

2. https://snap.stanford.edu/data/index.html
3. http://www.cnpameng.com/



(a) weibo4 (b) weibo4

Fig. 8. Time cost of algorithms when k_Q varies.



(a) facebook (b) weibo4

Fig. 10. Time cost of algorithms when η varies.

Varying k_Q. Fig. 8 presents the time cost and the number of $Sim(u, v|Q)$ computations for running each algorithm w. r.t. the same queries. In this study, we only focus on the *weibo4* dataset. Based on the results, we can see that *Index* is much faster than *Baseline* and *Improved* because it only calls for less number of the function $Sim(u, v|Q)$. Thus, its time cost is also lower. Note a user is a target user if his interests contain one of the query keywords, i.e., we use the "OR" query semantic in this work. When k_Q varies from 1 to 4, the numbers of calling for $Sim(u, v|Q)$ in *Improved* are about 719 times more than that of *Index*. The computation times of $Sim(u, v|Q)$ in *Baseline* depends on the size of the dataset and remains unchanged with the vary of k_Q. Compared with *Baseline* and *Improved*, *Index* reduces the time cost by about 101 times and 29 times respectively.

Varying τ and η. Figs. 9 and 10 present the time cost when we vary the users' interest similarity threshold τ and the spatial activation ratio η. Increasing τ may decrease the number of similarity users w.r.t. a user, thus it may lead to reducing the user's spatial influence spread. In contrast, the influence spread of a user may be weakened with an increase of η. More specifically, when τ varies from 0.01 to 0.5, *Baseline* consumes 386.12–438.40 seconds in *facebook*, and takes 7.69–8.02 hours in *weibo4*. The time cost of *Improved* increases from 19.75 seconds to 49.26 seconds in *facebook* and from 2.24 hours to 2.50 hours in *weibo4*. Furthermore, *Index* only takes 0.84 seconds and 269.91 seconds to terminate in the two datasets. When η varies from 0.001 to 0.1, the *baseline* takes 441.79–432.92 seconds in *facebook*. *Improved* terminates by consuming about 19.73–20.10 seconds, and *Index* stops by taking about 0.91–0.94 seconds.

5.3 Evaluation of Scalability

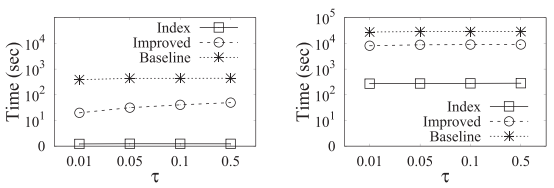
To demonstrate the scalability of our algorithms, we use *weibo4* and *Twitter* datasets. Based on the *weibo4* dataset, we generate fours subgraphs that contain the number of users from 1,000 to 50,000. Similarly, we produce another four subgraphs from *twitter* dataset by varying the corresponding number of users from 1,000 to 1×10^6. Fig. 11 presents the scalability of the algorithms over different size of vertices. It is noticed that the time cost of *Baseline* and *Improved* increases rapidly with the increase of dataset size. Compared with *Baseline* and *Improved*, *Index* consumes much less time to answer *HIM* queries on the same datasets. It is noticed that *Baseline* and *Improved* cannot get results in a valid time period once the number of users exceeds 1×10^5 in *Twitter* dataset, while *Index* can get the results in a valid period for all cases.

5.4 Evaluation of Effectiveness

As there is no ground truth to help verify the effectiveness of spatial influence diffusion, we develop a new metric to help recognize the impact of users' spatial influence spread in this section. We assume a seed set S can activate a set of target users, denoted as $\sigma(S)$, using our holistic influence diffusion model. Meanwhile, S can activate a set of users, denoted as $\sigma_1(S)$, using social influence diffusion model based on Yuchen *et al.* [18]. Thus, the set of users that are activated by S only using the spatial diffusion model is denoted as $\sigma_2(S) = \sigma(S) \setminus \sigma_1(S)$. As we discussed before, $weibo4_1$ and $weibo4_2$ contain the same set of users but with different posts in the two consecutive months. Our metric is designed based on the following motivation. If we run a campaign on $weibo4_1$ by running *HIM* query Q, then we can determine the minimum seed user set S w.r.t. Q and the set of influenced users $\sigma(S)$. And then, we check the same set of users $\sigma(S)$ in $weibo4_2$ and verify whether or not the Q-related keywords of users in $\sigma(S)$ have high *TF*IDF* scores in $weibo4_2$. The users in $\sigma(S)$ post their messages in $weibo4_2$, which are behind of $weibo4_1$ by one month. Therefore, if the observed score is high for a particular keyword, then it says that the users with the particular keyword in their new messages have been influenced by the keyword related campaign one month ago. Here, we randomly select 4 keywords as the campaigns for evaluating the effectiveness in both *weibo4* and *weibo10*.

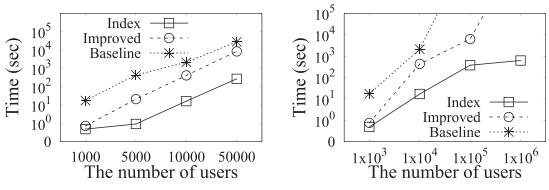
$$p_{ratio}(u) = \frac{\sum\{f_{tf*idf}(Q, w)|w \in T_u^2\}}{\sum\{f_{tf*idf}(Q, w)|w \in T_u^1\}}, \tag{9}$$

where T_u^1 represents the keywords in u's interest in $weibo4_1$, i.e., $T_u^1 \in weibo4_1$. And we also have $T_u^2 \in weibo4_2$. Similarly, we apply the above equation to test *weibo10* dataset.



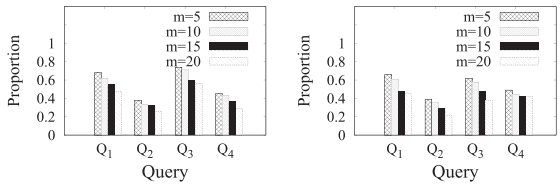
(a) facebook (b) weibo4

Fig. 9. Time cost of algorithms when τ varies.



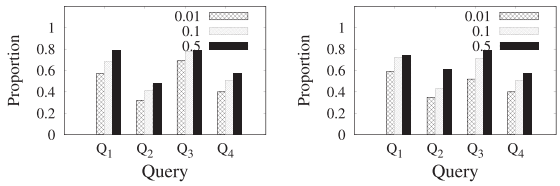
(a) weibo4 (b) twitter

Fig. 11. Scalability of algorithms when data size varies.



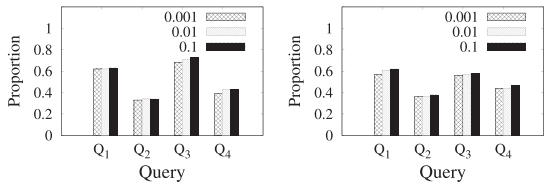
(a) weibo4 (b) weibo10

Fig. 12. Spatial influence impact for different queries and m.



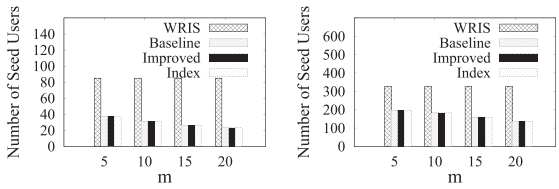
(a) weibo4 (b) weibo10

Fig. 13. Spatial influence impact when τ varies.



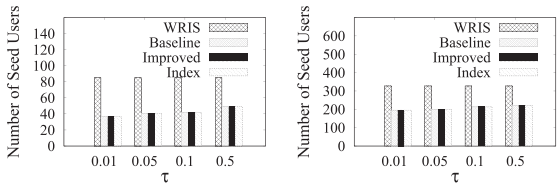
(a) weibo4 (b) weibo10

Fig. 14. Spatial influence impact when η varies.



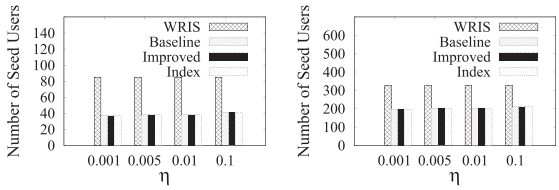
(a) facebook (b) weibo10

Fig. 15. The seed set size when *m* varies.



(a) facebook (b) weibo10

Fig. 16. The seed set size when τ varies.



(a) facebook (b) weibo10

Fig. 17. The seed set size when η varies.

In order to reduce the effect of local event to the result, for every user $u \in \{u|p_{ratio}(u) > 1 \wedge u \in \sigma_2(S)\}$, we further test whether it was influenced by local event. First, we choose a region R_u with user u as the center and the radius is equal to 20 km, the user in R_u is denoted as V_{R_u}, then we measure the influence impact of local events as

$$\frac{|\{v|p_{ratio}(v) > 1 \wedge v \in V_{R_u}\}|}{|V_{R_u}|}. \quad (10)$$

A parameter λ is used as the local events influence threshold. If the value of Equation (10) is no less than λ, we added user u into local events influence set LE. Thus, the *spatial influence impact* can be measured by

$$\frac{|\{u|p_{ratio}(u) > 1 \wedge u \in \sigma_2(S) \wedge u \notin LE\}|}{|\sigma_2(S)|}. \quad (11)$$

Varying m. Figs. 12a and 12b demonstrate the tracked spatial influence impact. The results show the difference on the spatial impact when we vary m. The precision is between 40-74 percent when m = 5. It varies in 25-56 percent when m is set as 20. The main reason is that the denominator $|\sigma_2(S)|$ becomes much larger when $m = 20$ than that of $m = 5$, but the nominator $|\{u|p_{ratio}(u) > 1 \wedge u \in \sigma_2(S)\}|$ may become smaller because only a few subset of users in $\sigma_2(S)$ have high *TF*IDF* score. It tells us that m is an important factor for system users to control the spatial influence spread of their advertisements.

Varying τ. Based on the same measurement using Equation (11), Fig. 13 presents the spatial influence impact of our *HIM* model by varying users' interest similarity threshold τ. From the results, we can see that the spatial influence increases apparently when we increase τ. It verifies our observation that the influence diffusion through physical connection clearly has a positive correlation with the user's interest similarity. But when τ reaches a specific large value (e.g., 0.5), the precision of spatial influence impact cannot be increased too much. This is because there are not many users who have the totally same interests, although our similarity metric also supports relevance, instead of exact matching keywords only.

Varying η. Fig. 14 shows the change when η varies. With the increase of η, the spatial influence impact is smoothly enhanced. But in most cases, the spatial influence impact is enhanced less by η than τ. For example, when η increases from 0.001 to 0.1, the spatial influence impact is only enhanced by 5 percent in Q_3 on *weibo4*.

5.5 Evaluation of *HIM*'s Benefit

We compare our *HIM* query with the *WRIS* method in [18]. *WRIS* selects the minimum number of seed users who can reach all target users based on the social influence of these seed users only. Our *HIM* query returns the minimum seed set based on our holistic influence diffusion. Figs. 15, 16, and 17 demonstrate the comparison of this two works when we vary the parameters. *WRIS* needs to choose about 85 seed users for reaching the target users in *facebook*, and about 327 seed users for reaching the target users in *weibo10*. For our HIM model-based methods, the number of selected seed users is 31 and 182 respectively. The varied parameters do not affect *WRIS*. But for our *HIM* query, the number of

seed users to be required varies with the change of parameters. When m increases, it needs a little fewer seed users. When τ increases, it needs more seed users due to the significant reduction of spatial influence impact. When η increases, it requires a little more seed users due to the slight decrease of spatial influence impact. The number of seed users in the *HIM* query is not affected by the proposed algorithms. So *Baseline*, *Improved* and *Index* have the almost same number of seed users in each test.

6 Related Work

The first line of relevant research are the influence maximization studies with the consideration of Location information. In [12], [16], the aim is to find the set of seed users whose influence spread is maximized within the given query region. Zhu *et al.* [31] studied the problem of location promotion by modeling the visiting probability of a user at a location based on his historic check-ins. Wang *et al.* [27] also studied a similar problem for answering a location point query. Li *et al.* [14] proposed a novel influence propagation model by learning the influence propagation of users across online social networks and the physical world. Their study is based on the overall statistic information, which can not be applied to capture the influence of individual users. Different from the above works, our *HIM* query investigates user-to-user influence by considering their social connections, spatial connections, and preference-similarities for targeted advertisement. The second line of research are relating to Topic-aware influence maximization works. Barbieri *et al.* [2] originally looked at social influence from topics perspective, and they proposed the Topic-aware influence cascade model. After that, there are many variants in this direction, e.g., [4], [15]. Besides, there are some research works concentrating on target user influence maximization. Li *et al.* [18] proposed the *KB-TIM* query and considered the influence on the target users. Similar to [18], our *HIM* query problem also targets to serve the targeted advertisement. But differently we consider the impact of dynamic spatial feature to the influence of users. There are other efforts to study influence maximization, e.g., diffusion models [10], efficient algorithms [26], and diversity [13]. We don't provide the details due to the limited space.

7 Conclusion

In this paper, we proposed a new research problem of *Holistic Influence Maximization* query that makes significant complementary to the traditional *influence maximization* problem, as well as brings additional benefits to many real applications in reality, e.g., event planning, advertisement placement, and crisis remedies, etc.. In *HIM* query, the user-to-user's influence model was formalized using three dimensional important factors: *social connection*, *spatial connection*, and *preference-based similarity connection*. Therefore, it is even harder to solve *HIM* query than the *IM* problem. To address this, we developed one baseline algorithm and one improved algorithm in this paper. In addition, a novel but simple spatial social index was devised to maintain the users' information in social networks and we also develop an index-based algorithm for further improving the efficiency of *HIM* query. The efficiency can be improved by about one or two orders of magnitude in our experimental evaluation using four datasets.

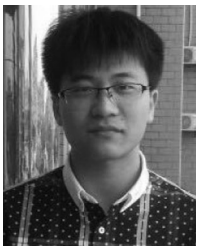
Acknowledgments

This work was mainly supported by ARC Linkage Project LP180100750, and partially supported by ARC Discovery Project DP190102443, the grant of the Research Grants Council of Hong Kong SAR, China, No. 14203618 and No. 14202919, and NSFC Grants 61772346 and National Key R&D Program of China 2018 YFB1004402.

References

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian, "Influence and correlation in social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 7–15.
[2] 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.
[3] C. Borgs, M. Brautbar, J. T. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2014, pp. 946–957.
[4] W. Chen, T. Lin, and C. Yang, "Efficient topic-aware influence maximization using preprocessing," *CoRR*, vol. abs/1403.0057, 2014. [Online]. Available: http://arxiv.org/abs/1403.0057
[5] 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. Discov. Data Mining*, 2010, pp. 1029–1038.
[6] J. Cheng, L. A. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, "Can cascades be predicted?" in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 925–936.
[7] P. S. Efraimidis and P. G. Spirakis, "Weighted random sampling with a reservoir," *Inf. Process. Lett.*, vol. 97, no. 5, pp. 181–185, 2006.
[8] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "Learning influence probabilities in social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 241–250.
[9] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "SIMPATH: An efficient algorithm for influence maximization under the linear threshold model," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 211–220.
[10] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.
[11] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A location-aware recommender system," in *Proc. IEEE 28th Int. Conf. Data Eng.*, 2012, pp. 450–461.
[12] 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.
[13] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Inf. Syst.*, vol. 92, 2020, Art. no. 101522.
[14] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.
[15] J. Li, C. Liu, J. X. Yu, Y. Chen, T. K. Sellis, and J. S. Culpepper, "Personalized influential topic search via social network summarization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1820–1834, Jul. 2016.
[16] J. Li, T. Sellis, J. S. Culpepper, Z. He, C. Liu, and J. Wang, "Geo-social influence spanning maximization," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1653–1666, Aug. 2017.
[17] J. Li, X. Wang, K. Deng, X. Yang, T. Sellis, and J. X. Yu, "Most influential community search over large social networks," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 871–882.
[18] Y. Li, D. Zhang, and K. Tan, "Real-time targeted influence maximization for online advertisements," *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1070–1081, 2015.
[19] B. Liu, G. Cong, D. Xu, and Y. Zeng, "Time constrained influence maximization in social networks," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 439–448.
[20] C. Long and R. C. Wong, "Viral marketing for dedicated customers," *Inf. Syst.*, vol. 46, pp. 1–23, 2014.

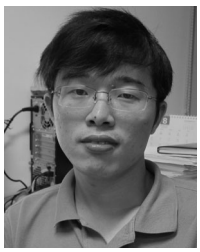
[21] B. Lucier, J. Oren, and Y. Singer, "Influence at scale: Distributed computation of complex contagion in networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 735–744.
[22] S. A. Myers, C. Zhu, and J. Leskovec, "Information diffusion and external influence in networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 33–41.
[23] N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi, "Dynamic influence analysis in evolving networks," *Proc. VLDB Endowment*, vol. 9, no. 12, pp. 1077–1088, 2016.
[24] C. Song, W. Hsu, and M. Lee, "Targeted influence maximization in social networks," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1683–1692.
[25] I. Stanoi, D. Agrawal, and A. El Abbadi , "Reverse nearest neighbor queries for dynamic databases," *ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, pp. 44–53, 2000.
[26] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 75–86.
[27] X. Wang, Y. Zhang, W. Zhang, and X. Lin, "Distance-aware influence maximization in geo-social network," in *Proc. IEEE 32nd Int. Conf. Data Eng.*, 2016, pp. 1–12.
[28] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.
[29] S. Yang, M. A. Cheema, X. Lin, and W. Wang, "Reverse k nearest neighbors query processing: Experiments and analysis," *Proc. VLDB Endowment*, vol. 8, pp. 605–616, 2015.
[30] T. Zhou, J. Cao, B. Liu, S. Xu, Z. Zhu, and J. Luo, "Location-based influence maximization in social networks," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1211–1220.
[31] W. Zhu, W. Peng, L. Chen, K. Zheng, and X. Zhou, "Modeling user mobility for location promotion in location-based social networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1573–1582.



Taotao Cai received the ME degree in computer science from Shenzhen University, China in 2016. He is currently working toward the PhD degree at Deakin University, Australia. His research interests include algorithmic aspects of geo-social network analysis and database query processing and optimization

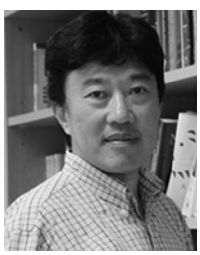
Jianxin Li received the PhD degree in computer science from the Swinburne University of Technology, Australia, in 2009. He is an associate professor in data science in the School of Information Technology, Deakin University, Australia. His research interests include graph database query processing and optimization, social network analytics and computing, complex network representation learning, and personalized online learning analytics. He is also a grant assessor in Australia Research Council Discovery Programs and Linkage Programs, and serves as invited reviewers for many top journals and program committee members in many top conferences.

Ajmal Mian is an associate professor of computer science at The University of Western Australia. He has received two prestigious national fellowships and seven competitive grants from the Australian Research Council and the National Health and Medical Research Council. His research interests include action recognition, 3D shape analysis, and machine learning.



Rong-Hua Li received the PhD degree from the Chinese University of Hong Kong, Hong Kong, in 2013. He is currently an associate professor at the Beijing Institute of Technology, Beijing, China. His research interests include graph data management and mining, social network analysis, graph computation systems, and graph-based machine learning.

Timos Sellis (Fellow, IEEE) received the PhD degree in computer science from the University of California, Berkeley, Berkeley, California, in 1996. He is a professor with the Swinburne University of Technology, Australia. Till the end of 2012, he was the director of the Institute for 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, data integration, and spatio-temporal database systems. He is a fellow of the ACM.



Jeffrey Xu Yu received the PhD degrees in computer science from the University of Tsukuba, Japan, in 1990. He has held teaching positions at the Institute of Information Sciences and Electronics, University of Tsukuba, Japan, and at the Department of Computer Science, Australian National University, Australia. Currently, he is a professor with the Department of Systems Engineering and Engineering Management, the Chinese University of Hong Kong, Hong Kong. He is serving as a VLDB Journal Editorial Board member. His current research interests include graph database, graph mining, keyword search in relational databases, and social network analysis.