Maximizing the influence of bichromatic reverse $k$ nearest neighbors in geo-social networks

Pengfei Jin$^1$ · Lu Chen$^1$ · Yunjun Gao$^1$ · Xueqin Chang$^1$ · Zhanyu Liu$^1$ · Shu Shen$^2$ · Christian S. Jensen$^3$

Received: 18 July 2022 / Revised: 12 August 2022 / Accepted: 18 August 2022 / Published online: 3 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Geo-social networks offer opportunities for the marketing and promotion of geo-located services. In this setting, we explore a new problem, called Maximizing the Influence of Bichromatic Reverse $k$ Nearest Neighbors (MaxInfBR$k$NN). The objective is to find a set of points of interest (POIs), which are geo-textually and socially relevant to social influencers who are expected to largely promote the POIs online. In other words, the problem aims to detect an optimal set of POIs with the largest word-of-mouth (WOM) marketing potential. This functionality is useful in various real-life applications, including social advertising, location-based viral marketing, and personalized POI recommendation. However, solving MaxInfBR$k$NN with theoretical guarantees is challenging because of the prohibitive overheads on BR$k$NN retrieval in geo-social networks, and the NP and #P-hardness of finding the optimal POI set. To achieve practical solutions, we present a framework with carefully designed indexes, efficient batch BR$k$NN processing algorithms, and alternative POI selection policies that support both approximate and heuristic solutions. Extensive experiments on real and synthetic datasets demonstrate the good performance of our proposed methods.

Keywords Geo-social networks · Bichromatic reverse $k$ nearest neighbors · Social influencers · POI recommendation · Algorithms

1 Introduction

The problem of bichromatic reverse $k$ nearest neighbor (BR$k$NN) search and maximizing BR$k$NN (MaxBR$k$NN) have received attention due to their importance in a wide range of applications [12, 26, 50]. Given two sets $P$ and $U$ representing points of interest (POIs) and users, respectively, the BR$k$NN query is issued for a data point $p \in P$, in order to find all users $u \in U$ having $p$ as one of their $k$ nearest neighbors under a given distance definition [26]. The set of BR$k$NN results is also referred as the influence set of $p$, and can be utilized for finding prospective customers [55]. Based on BR$k$NN, the MaxBR$k$NN problem and
corresponding variants are proposed to maximize the size of BR\(k\)NN results via optimal location selection \([50]\) or geo-textual tags recommendation \([11, 12]\), and hence target at location-aware marketing and promotion for geo-located services.

The increasing popularity of geo-social networks offers more opportunities for location-based marketing. By leveraging the word-of-mouth effect, sellers can advertise geo-tagged services online to attract more customers. In social media platforms, influencers, also called cyber-celebrities, play an important role in information dissemination. It is reported that in the USA, more than half of social media users prefer purchasing products or services recommended by the influencers they follow, and about 66\% of users believe recommendations by influencers are credible\(^1\). Although influencers and the underlying influence propagation between users exist in the real scenarios, existing BR\(k\)NN and MaxBR\(k\)NN studies \([12, 50]\) rarely take these into account. Rather, they assign equal importance to all users in BR\(k\)NNs, and they focus on maximizing the size of BR\(k\)NNs. The following example illustrates this situation.

**Example 1** Figure 1a depicts six users (circles) and three convenience stores (triangles), where the dotted line between a user \(u\) and a store \(p\) means that \(u\) has \(p\) as one of their \(k\) nearest neighbors. Figure 1b shows a social network where edges represent the social links between users, with edge labels indicating the probability of one user influencing another user. Suppose stores \(p_1\) and \(p_2\) belonging to the same chain, and the manager plans to select the optimal one to launch a marketing campaign. A traditional MaxBR\(k\)NN would return \(p_2\), because \(p_2\) has the largest BR\(k\)NN result set. However, as shown in Figure 1b, the users with \(p_2\) as nearest store exert less influence and cannot propagate their influence to other users; thus, the influence is constrained to only three users. To fully enhance the online influence, a better choice may be \(p_1\), as users \(u_1\) and \(u_2\) that are attracted by \(p_1\) embody substantial influence, and they may be able to spread their influence throughout the social network.

![Figure 1](image.png)

We also note that existing related studies \([11, 12, 15]\) aim at enhancing the BR\(k\)NN results for only a single target. In contrast, for sellers with multiple stores or services, a combined promotion for multiple targets should be preferable over a single promotion target. We thus formulate a problem in geo-social networks called **Maximizing the Influence of Bichromatic Reverse \(k\) Nearest Neighbors (MaxInfBR\(k\)NN)**. Given a query set \(\mathcal{P}_c \subseteq \mathcal{P}\), and two integers \(b\) and \(k\), MaxInfBR\(k\)NN aims to find an optimal

\(^1\) [https://edition.cnn.com/business/newsfeeds/globenewswire/7812666.html](https://edition.cnn.com/business/newsfeeds/globenewswire/7812666.html)
subset $P_s$ of $P_c$, such that $|P_s| = b$ and the elements in $P_s$ are highly relevant (i.e., become top-$k$ results) to influencers who can effectively spread their influence online. Here, $P$ is the ground set of POIs in the dataset. $P_c \subseteq P$ is the candidate set with multiple target POIs. $P_s$ is the query result selected from $P_c$. For example, in real scenarios, $P_c$ may include the chains belonging to the same company (i.e., Walmart, Starbucks), and the manager plans to select $b$ chains from $P_c$ for promotion, such that the promoted POI set (i.e., $P_s$) embody the largest social influence. We assume that users and POIs are located in a road network, since movement in real-life settings is constrained to a spatial road network [55]. When measuring similarities between users and POIs, we consider both geo-textual and social relevance scores, as social media users are more likely to visit nearby relevant places that are also favored by their friends [15]. Advertisements in online platforms normally have the limited space or budgets, meaning that $b (= |P_s| \ll |P_c|)$ should be reasonably small.

Answering MaxInfBR$k$NN queries poses new challenges. A straightforward solution could be to first retrieve BR$k$NN results for each $p \in P_s$, then enumerate all the possible size-$b$ POI combinations and return the optimal one whose combined BR$k$NN results have the largest social influence. Although this approach is intuitive, it might be infeasible in practice. The major challenge comes from the prohibitive BR$k$NN retrieval costs in geo-social networks. Existing solutions [55] tackled this with a time complexity of $O(|P| \cdot |U|^2)$, which fails to scale to larger datasets. Moreover, although exact BR$k$NN results can be computed efficiently, finding the optimal POI set whose BR$k$NNs have the largest influence is NP-hard and also inherits #P-hardness in computing social influence [24]. Fortunately, this process can be accelerated by sampling-based techniques [17, 43], but it is still costly and not sufficiently robust across different inputs (confirmed by experiments in Section 8).

Motivated by the potential benefits of the MaxInfBR$k$NN query and the lack of effective solutions, this paper makes the following contributions:

- We formalize the MaxInfBR$k$NN query in geo-social networks. To the best of our knowledge, this is the first attempt to systematically tackle this problem.
- We prove that the problem is NP-hard, and we present a non-trivial baseline solution with theoretical guarantees.
- We develop a batch processing algorithm to retrieve BR$k$NNs for multiple geo-social objects, the complexity is $O(\xi_1 \cdot |U| + \xi_2)$ ($\xi_1(\xi_2) \ll |P|$), which ensures the scalability of our proposals and also is a contribution to existing BR$k$NN studies.
- We propose robust and alternative POI selection policies to answer MaxInfBR$k$NN either approximately or heuristically without excessive sampling costs.
- We report on a comprehensive empirical study to offer insight into the effectiveness and efficiency of our proposed indexes and algorithms.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 covers preliminaries and formalizes our problem. Section 4 presents a non-trivial baseline solution. Section 5 gives a high-level overview of our framework. Section 6 covers our indexing schemes and algorithms for batch BR$k$NN processing. Section 7 presents alternative POI selection policies to efficiently answer MaxInfBR$k$NN queries. We report on an experimental study in Section 8. Finally, Section 9 concludes the paper, and provides future work suggestions.
2 Related work

This section briefly covers studies related to our problem, including BR$k$NN and MaxBR$k$NN queries, influence maximization, and location-based data analytics.

2.1 BR$k$NN and MaxBR$k$NN Queries

The bichromatic reverse $k$ nearest neighbor (BR$k$NN) query was first studied by Korn et al. [26]. Given two sets $U$ and $P$ of data points, and an object $p \in P$, the query finds all points in $U$ that have $p$ as one of their $k$ nearest neighbors in $P$. Since then, this query has been explored in extensive application settings, including, for example, the $R^k$NN in large graphs [53], the Reverse Spatial and Textual $k$ Nearest Neighbor (RST$k$NN) search [28, 29], and the Reverse Top-$k$ Geo-Social Keyword Query (R$k$ GSKQ) [55]. Recently, inspired by profile-based marketing [50], the MaxBR$k$NN query that maximizes the result size of BR$k$NNs was studied. Considerable efforts have been made, e.g., MaxBR$k$NN querying acceleration [9], MaxBRST$k$NN for geo-textual data [12, 15], and MaxBR$k$NN for streaming geo-data [30], to name just a few. These studies all assign equal importance to each data point, and focus on maximizing the result size. However, different data points in BR$k$NNs may exert different influence. This motivates our problem. Huang et al. [21] consider social influence of BR$k$NNs, but only find a single optimal location, while our problem aims to find an optimal set of geo-social objects, which is NP-hard. Moreover, the similarity distance defined by by Huang et al. [21] is the Euclidean distance, whereas our similarity notion considers both geo-textual and social relevance, which is more realistic.

2.2 Influence maximization

Influence maximization (IM) was first studied by Domingos and Richardson [13, 35]. The goal is to find a set of users called seeds to trigger the largest expected influence propagation in social networks. Kempe et al. [24] formulated the problem as a discrete optimization problem, proved its NP-hardness [24], and proposed an algorithm with a $(1 - 1/e)$-approximation ratio by Monte-Carlo simulation. Since then, substantial research has been devoted to developing more efficient and scalable IM algorithms [17, 43, 44, 46]. Among them, the reverse influence sampling (RIS) techniques (first presented in [5] and subsequently optimized [17, 43, 46]) are widely considered as the state-of-the-art in solving the IM problem with theoretical guarantees. Recently, RIS has also been extended to address many IM variants [4, 20, 31]. In addition, the topics of location-aware IM have received considerable recent attention [10, 27, 48]. However, all of the work mentioned above only consider that users can be compulsively selected as seeds regardless of their personal preferences. To bridge this gap, our problem focuses on finding multiple target POIs that are attractive to social influencers, who can contribute to online marketing campaigns. Our problem is query-dependent and thus is more flexible in personalized applications.

2.3 Location-based data analytics

Driven by the advances in geo-positioning devices and the mobile Internet, the location-based data analytics play important roles in daily life. Many fundamental techniques and
topics have been studied in this scope, which includes (but not only includes) spatial keyword search [6–8, 51], POI recommendation [18, 19, 34], location mining via geo-social data [39, 42, 49], and location-aware querying in road networks [38, 40, 41, 52, 57], to name but a few. The problem of MaxInfBR_kNN can be regarded as a variant of spatial keyword search. Meanwhile, it is also one of the location-aware query types in road networks, which considers both geo-textual and social information and returns personalized results. This functionality is also applicable to POI recommender systems, with the focus of retrieving a group of POIs for sellers. Although our problem is related to all above topics, their technical details are different, hence existing solutions cannot directly solve our problem.

### 3 Preliminaries

Table 1 lists the frequently used notations. In our problem, a geo-social network is a heterogeneous network as shown in Figure 2a. Specifically, a geo-social network is composed by a road network $G_r = (V_r, E_r)$ and a social network $G_s = (V_s, E_s)$, where $V_r$ ($V_s$) and $E_r$ ($E_s$) denote the vertex set and the edge set respectively. Two types of geo-social

| Notations | Description |
|-----------|-------------|
| $G_r = (V_r, E_r)$ | a spatial road network |
| $G_s = (V_s, E_s)$ | an online social network |
| $\mathcal{P}(U)$ | a set of POIs (users) located on $G_r$ |
| $dist(p,u)$ | the shortest path distance from a POI $p$ to a user $u$ |
| $F_{GST}(u,p)$ | the geo-social and textual similarity score of $p$ to $u$ |
| $S_t(u)$ | the top-$k$ (kNN) result set of $u$ |
| $S_t(p)$ | the bichromatic reverse kNN result set of $p$ (the influence set of $p$) |
| $NV_D$ | the network voronoi diagram generated for the POIs covering keyword $t$ |
| $\mathcal{P}_c$ | the set of candidate POIs which is given as a query input |
| $\mathcal{P}_j$ | the optimal POI set selected from $\mathcal{P}_c$ which is the query result |
| $I_P(\mathcal{P}_i)$ | the influence of a POI set $\mathcal{P}_i$ |

Figure 2 A toy example of geo-social network
and textual objects are located in $\mathcal{G}_t$, i.e., a set $\mathcal{P}$ of POIs and a set $\mathcal{U}$ of users that have reviews on POIs and correspond to the nodes in $\mathcal{G}_t$. Each POI $p \in \mathcal{P}$ is a triple $(\text{loc}, \text{key}, \text{CK})$, where $p.\text{loc} = (v_{NN}.\text{dis})$ is a geo-spatial position in $\mathcal{G}_r$, given by the nearest vertex $v_{NN}$ of $p$ in $\mathcal{G}_t$, and a distance $\text{dis}$ to that vertex; $p.\text{key}$ is a set of keywords coupled with their occurrence frequencies in the text of $p$ [7]; $p.\text{CK}$ is a set of IDs of users who have checked into $p$. Each user $u \in \mathcal{U}$ is denoted by a triple $(\text{loc}, \text{key}, F(u))$, where $u.\text{loc}$ is the location descriptor similar as $p.\text{loc}$, $u.\text{key}$ is a set of keywords capturing the users’ preferences, and $F(u)$ is the friend set of $u$ in $\mathcal{G}_t$. The textual and social information of POIs (users) are shown in Table 2. Note that $u_9$ to $u_{12}$ in Figure 2 have no keywords. Thus they are not shown in Table 2.

**Definition 1 (Top-k Geo-Social Keyword (TkGSK) queries).** Given a POI set $\mathcal{P}$ and a user $u$, a Tk GSK query is issued by $u$ to find a set $S_k(u)$ with $k$ POIs $p$ in $\mathcal{P}$, which are most relevant to $u$ based on the scoring function as below:

$$F_{GST}(u, p) = \frac{\alpha \cdot f_s(u, p) + (1 - \alpha) \cdot f_t(u, p)}{f_s(u, p)}$$

In Eq. 1, $f_s(u, p)$ denotes the geographical proximity computed by the shortest path distance between $u$ and $p$ in $\mathcal{G}_t$, $f_t(u, p)$ represents the textual similarity based on the TF-IDF metric [11, 37], and $f_s(u, p)$ is the social relevance computed by $f_s(u, p) = \frac{\sum p \in \text{CK} (F(u) \cap p.\text{CK})}{|F(u)|}$, which follows standard social score formulations [2]. The parameter $\alpha \in [0, 1]$ balances the importance of social relevance and textual similarity. Here, a large score of $p$ means that $p$ has high relevance to $u$. Note that Eq. 1 is formulated as a ratio rather than as a linear function which is common in existing studies [2, 55]. This is done to avoid expensive normalization of the geographical score with prior knowledge [36] (i.e., the largest shortest path distance). Nonetheless, our methods can easily be extended to support the linear function as well.

**Definition 2 (Bichromatic Reverse k Nearest Neighbor (BRkNN) in the geo-social network).** Given a set $\mathcal{P}$ of POIs and a set $\mathcal{U}$ of users, in a geo-social network, a BRkNN query is issued for a POI $p \in \mathcal{P}$, and returns all the users $u \in \mathcal{U}$ with $p$ in their Tk GSK query results. Thus, if the result is denoted as $S_k(p)$, we have $S_k(p) = \{u | u \in \mathcal{U} \land p \in S_k(u)\}$.
Example 2 In Figure 2a, the T2GSK query result set \( S(u_1) \) for \( u_1 \) is \( \{p_1,p_2\} \), as \( p_1 \) and \( p_2 \) are most relevant to \( u_1 \) among all \( p \in P \) according to Eq. 1. Similarly, \( S(u_2) = \{p_1,p_2\} \) and \( S(u_3) = \{p_1,p_3\} \). Thus, \( S_i(p_1) = \{u_1,u_2,u_3\} \).

For a POI set \( P_i \), we denote \( S_i(P_i) \) as the union of BR\( k \)NN results for all \( p \in P_i \), i.e., \( S_i(P_i) = \bigcup_{p \in P_i} S_i(p) \). As the POIs in \( P_i \) are highly relevant to the users in \( S_i(P_i) \), these users tend to be influenced/attracked by \( P_i \), and become their potential propagators. For simplicity, we also refer to the BR\( k \)NNs as potential users. We next introduce the Independent Cascade (IC) model [24], a classic and dominant influence propagation model, to formulate the influence of different users in BR\( k \)NNs.

Definition 3 (IC Model) [24]. Given a social network \( G_s = (V_s,E_s) \), the IC model assigns a weight \( w(u,v) \in [0,1] \) to each edge \((u,v) \in E_s \) that denotes the probability that \( v \) can be influenced by \( u \). The influence propagation over \( G_s \) is modeled as an iterative stochastic process. Initially, a set of users called seeds (denoted as \( S \) ) are influenced. Then, seed users further spread their influence through the social network following randomized rules. Specifically, in each iteration, when a user \( u \) is newly influenced/activated, the user has a single chance to activate each of inactive friend \( v \) with probability \( w(u,v) \). Influenced users remain in this state until the end. The propagation process proceeds until no users in \( G_s \) can be further influenced.

Remark 1 Note that, due to the space limitation, we only discuss our problem under the IC model in this paper, as the IC model is considered as the most standard diffusion model [32] and discussed in the first priority in many latest researches [16, 20, 32]. We leave it as future work to further generalize our problem to other popular diffusion models (i.e., the LT model [24], continuous-time IC model [46]).

Definition 4 (Influence of users (POIs)). Let \( I(S) \) be the total number of users influenced by the propagation from \( S \). Due to the randomness in the propagation process, the influence \( I(S) \) of \( S \) is defined as the expected number of influenced users over all the possible propagation instances i.e., \( I(S) = E(I(S)) \) [24]. Base on this, the influence of a POI set \( P_i \) (i.e., \( I_P(P_i) \)) is evaluated by \( I(I(S_i(P_i))) \), which is the expected influence aroused by \( S_i(P_i) \), when the users in BR\( k \)NNs of \( P_i \) seed the influence propagation process under the IC model.

Definition 5 (MaxInfBR\( k \)NN). Given a query set \( P_c \) (\( P_c \subseteq P \) ) and a budget value \( b \) (\( b \leq |P_c| \) ), the MaxInfBR\( k \)NN problem is to find a set \( P_s \subseteq P_c \), such that \( |P_s| = b \) and the influence propagation achieved by the BR\( k \)NNs of \( P_s \) is the largest over all size-\( b \) subsets of \( P_c \). Formally, \( P_s = \arg \max_{P_s \subseteq P_c, |P_s| = b} I_P(P_s) \).

Example 3 In Figure 2, assume that a company owns three stores \( P_c = \{p_1,p_3,p_5\} \). When \( k = 2 \), the potential users for each \( p \in P_c \) are \( S_i(p_1) = \{u_1,u_2,u_3\} \), \( S_i(p_3) = \{u_3,u_4,u_5\} \), and \( S_i(p_5) = \{u_6,u_7\} \). For simplicity, we set the weight of each edge in \( G_s \) to 1. Given \( b = 2 \), the optimal size-2 POI set selected from \( P_c \) is \( \{p_1,p_3\} \), whose influence is 10, which is the total number of influenced users in \( G_s \), including potential users (i.e., \( u_1,u_2,u_3,u_4,u_5 \) and \( u_6 \) ) of \( \{p_1,p_3\} \) and the users (i.e., \( u_6,u_9,u_{10},u_{11},u_{12} \) ) further activated by the potential users via influence propagation.
At the first glance, the definition of MaxInfBR\(_k\)NN queries may resemble to existing work on reverse \(k\)NN queries \([28, 55]\) and some BR\(_k\)NN query variants \([12, 50]\). However, the novelty of MaxInfBR\(_k\)NN queries is still substantial over these work in terms of two aspects. First, the MaxInfBR\(_k\)NN query takes into account the influence difference between users in BR\(_k\)NN, and detects an optimal set of POI whose BR\(_k\)NNs exert the largest social influence. In contrast, existing works on reverse \(k\)NN queries \([28, 55]\) only retrieve query results without any optimization towards the results. Although there exist some query variants (i.e., MaxBR\(_k\)NNs \([50]\) and MaxBRST\(_k\)NN \([12]\)), they only focused on optimizing the size of BR\(_k\)NNs, not their influence. Hence, the functionalities of these queries may be limited and inferior to MaxInfBR\(_k\)NN queries in viral marketing scenarios. Secondly, the fundamental step in solving a MaxInfBR\(_k\)NN query is retrieving BR\(_k\)NNs for a batch of query objects. But how to efficiently process batch BR\(_k\)NN queries is still an open problem, which makes our problem more challenging compared to existing ones \([12, 28, 55]\).

**Lemma 1** The exact solution to MaxInfBR\(_k\)NN is NP-hard.

**Proof** The problem of MaxInfBR\(_k\)NN can be reduced from a well-known NP-hard problem, the Maximum Coverage (MC) problem \([14]\). Given a collection of subsets \(\mathcal{S} = \{S_1, S_2, ..., S_m\}\) of a ground set \(\mathcal{G} = \{e_1, e_2, ..., e_n\}\), and a positive integer \(b\), the MC problem aims to find \(\mathcal{S}' \subseteq \mathcal{S}\), such that \(|\mathcal{S}'| = b\) and \(\mathcal{S}'\) covers the largest number of distinct elements in \(\mathcal{G}\). The problem of MaxInfBR\(_k\)NN is a generalization of MC. If each user in \(S_r(p)\) has no social links in \(G_s\) (a.k.a. cold start users), then the problem of solving MaxInfBR\(_k\)NN becomes equivalent to solving the MC problem, with \(S_r(p)\) of each \(p \in \mathcal{P}_c\) corresponding to each subset in \(\mathcal{S}\) defined by the MC problem. Therefore, the problem of MaxInfBR\(_k\)NN is at least as hard as the MC problem, and thus MaxInfBR\(_k\)NN is NP-hard. □

4 Baseline solution

MaxInfBR\(_k\)NN is NP-hard, and also inherits #P-hardness in exactly computing social influence \([24]\). To avoid brute force search, we can theoretically approximate the optimal solution based on the following lemma.

**Lemma 2** The objective function \(I_p(\cdot)\) evaluating the influence of a set of POIs is a submodular function, which satisfies \((I_p(\mathcal{P}_j \cup \{p\}) - I_p(\mathcal{P}_j)) \leq (I_p(\mathcal{P}_i \cup \{p\}) - I_p(\mathcal{P}_i))\) when \(\mathcal{P}_j \supseteq \mathcal{P}_i\).

**Proof** The detailed proof of this lemma are presented in our extended version \([22]\).

Since submodular optimization problems can be solved with an approximation ratio no worse than \(1 - 1/e \approx 0.632\) \([14]\), hence we present a non-trivial baseline (denoted as BA) to solve the problem with existing techniques. The workflow of BA includes major components which are summarized as follows:

**BR\(_k\)NNs retrieval** First, we perform the state-of-the-art GIM-Tree based algorithm \([23, 55]\) to retrieve the exact BR\(_k\)NN results for each \(p \in \mathcal{P}_c\). Then, we add a directed edge with weight 1 from each POI \(p \in \mathcal{P}_c\) to a user \(u\) if \(u \in S_r(p)\). This yields a heterogeneous graph...
\( G_H \), whose nodes contain both users and POIs. To avoid unnecessary computations, all the users in \( G_H \) are removed when they cannot be reached by any POI in \( P_c \).

**Reverse influence sampling** Second, we extend the state-of-the-art reverse influence sampling (RIS) technique [17, 43] to support POI selection. The core idea of RIS is to generate random reverse reachable (RR) sets [5] to estimate the social influence. However, our RR set generation differs from previous methods [17, 43], since we only generate RR sets particularly for POIs. More specifically, we implement RR set generation by uniformly sampling a user \( u_i \) from \( G_H \) and performing stochastic reverse breadth first search (BFS) from \( u_i \) [46]. The difference is that our BFS is conducted on \( G_H \) rather than \( G_s \) and only adds POIs in \( P_c \) reached by \( u_i \) to the RR set. More optimizations for RR set generation are available in [17].

Following previous studies [17, 43], we generate two collections of RR sets \( R_1 \) and \( R_2 \) in a sequential manner and then utilize \( R_1 \) and \( R_2 \) to determine selected POIs.

**POI selection and qualification** Whenever \( |R_1| = |R_2| = 2^{i} \), we apply the greedy algorithm to select \( b \) POIs from \( P_c \) covering the largest number of RR sets in \( R_1 \). Let \( P^*_b \) be the POI set selected by the greedy algorithm. Then, we utilize \( R_2 \) to determine an influence lower bound \( \Pi^\text{−}_b(P^*_b) \) on \( \Pi_b(P^*_b) \). Let \( P^\text{opt}_b \) denote the POI set with the largest influence among all size-\( b \) subsets of \( P_c \). Next, we further use \( R_1 \) to derive an influence upper bound \( \Pi^\text{+}_b(P^\text{opt}_b) \) on \( \Pi_b(P^\text{opt}_b) \). According to \( \Pi^\text{−}_b(P^*_b) \) and \( \Pi^\text{+}_b(P^\text{opt}_b) \), the worst approximation ratio of \( P^*_b \) is computed by \( \Pi^\text{−}_b(P^*_b)/\Pi^\text{+}_b(P^\text{opt}_b) \). Given an error threshold \( \epsilon \), if the derived approximation ratio is no less than \( 1 - 1/e - \epsilon \), \( P^*_b \) is returned. Otherwise, we repeat procedures 2 and 3 until \( P^*_b \) is qualified. The bounds formulation of both \( \Pi^\text{−}_b(P^*_b) \) and \( \Pi^\text{+}_b(P^\text{opt}_b) \) are covered in Lemma 3.

**Lemma 3** Given \( R_1, R_2, P^*_b \), and \( P^\text{opt}_b \) defined as above, let \( \Lambda_{R_2}(P^*_b) \) be the number of RR sets covered by \( P^*_b \) in \( R_2 \) (the coverage of \( P^*_b \) in \( P^\text{opt}_b \) ), and \( \Lambda_{R_1}(P^\text{opt}_b) \) be the upper bound of the coverage of \( P^\text{opt}_b \) in \( R_1 \). Then, we have

\[
\Pi^\text{−}_b(P^*_b) = \left( \sqrt{\Lambda_{R_2}(P^*_b)} + \frac{2\eta_l}{9} - \sqrt{\frac{\eta_l}{2}} \right)^2 \cdot \frac{|V_1|}{|R_2|} \tag{2}
\]

\[
\Pi^\text{+}_b(P^\text{opt}_b) = \left( \sqrt{\Lambda_{R_1}(P^\text{opt}_b)} + \frac{\eta_u}{2} + \sqrt{\frac{\eta_u}{2}} \right)^2 \cdot \frac{|V_1|}{|R_1|} \tag{3}
\]

where \( \eta_l = \ln(1/\delta_l) \) (\( \eta_u = \ln(1/\delta_u) \)), \( \delta_l \) (\( \delta_u \)) is the probability when the lower (upper) bound estimation in (2) (3) fails. In [17, 43], \( \delta_l = \delta_u = 1/2 \cdot \delta \), with \( \delta \) as the total error probability whose value is tunable according to users’ requirements.

**Proof** The lemma is extended from Lemmas 4.2 and 4.3 in [43] proposed by Tang et al.

**Discussion** Although the baseline solution can solve our problem with theoretical guarantees, it is still infeasible in practice due to three reasons. i) The time complexity of existing BR\&NNs retrieval in geo-social networks is \( O(|P| \cdot |U|^2) \) [55], which fails to scale to large geo-social networks. ii) The index (i.e., the GIM-Tree) used in the baseline is a memory consuming index. This index has to maintain two large matrices [54, 55] for social...
relevance computation, whose space cost of is $O(|\mathcal{P}| \cdot |\mathcal{U}|)$, which might yield prohibitive indexing costs. iii) Directly using RIS to evaluate the influence of POIs would lead to unstable running performance, especially when $b$ is small in real-life settings (as indicated by our empirical results). To ensure the accuracy, more RR sets have to be generated by BA to qualify the results, which results in expensive sampling costs and lower query efficiency.

5 The overview of our framework

To overcome the deficiencies in BA, we propose a framework as shown in Figure 3. This framework is composed of two levels, each including technical contributions. Specifically, the first level in our framework is to support efficient and scalable batch BR$k$NN retrieval (to be introduced in Section 6). Retrieving BR$k$NNs for multiple POIs is a prerequisite for solving MaxInfBR$k$NN with approximation guarantees, but is still very costly in practice. Hence, the techniques in this level ensure the scalability of our proposals. The second level offers strategies to effectively evaluate the influence of BR$k$NNs and select influential POIs. Two alternative solutions are presented, including an approximation solution (denoted as AP) that returns theoretically guaranteed results (to be elaborated on in Section 7.2), and a heuristic solution (denoted as HE) that solves the problem more efficiently without theoretical guarantees while still returning highly qualified results in practice (to be covered in Section 7.3).

6 Batch BR$k$NN processing

We first introduce a new index scheme with accompanying pruning techniques. Based on these, we develop efficient and scalable algorithms for batch BR$k$NN processing.

6.1 Indexing scheme

Given that social information in geo-social networks is often dynamic and requires extra update costs if indexed, we index only the geo-textual information which is relatively stable in real scenarios. Our index called IG-NVD, is a variant of the inverted G-Tree [56] that incorporates Network Voronoi Diagrams (NVD) [25].

As shown in Figure 4, the backbone of IG-NVD is a G-Tree [56], which recursively partitions a road network into sub-networks. The root node of a G-Tree represents the whole road network $\mathcal{G}_r$, and each sub-network $\mathcal{G}_r^i = (\mathbf{V}_r^i, \mathbf{E}_r^i, \mathbf{R}_r^i, \mathbf{P}_r^i, \mathbf{U}_r^i)$ is represented by a
tree node $N_r$. Here, $V_r$, $E_r$, and $B_r$ denote the sets of vertices, edges, and border vertices in $G_r$, and $P_r$ $(U_r)$ is the set of POIs (users) located in $G_r$. Each tree node $N_r$ is associated with an inverted file and several NVDs. The inverted file of each node (e.g., $InvF_1$ of $N_1$ in Figure 4b) indexes the textual information of users and POIs in the child nodes, which facilitates keyword-aware road network access.

The NVDs are constructed by dividing the road network based on the locations of POIs in the road network $G_r$. For a set $P'$ of POIs covering the keyword $t$, the NVD for $P'$ is a data structure dividing $G_r$ into disjoint partitions, such that each partition corresponds to a specific POI $p \in P'$ and the set of vertices in $G_r$ having $p$ as their nearest neighbor (NN). Let $NV P'$ be the NVD built for $P'$. NV $P'$ can be abstracted as an adjacency graph, whose node $NV P(p_i)$ represents the network voronoi partition (NVP) for each $p \in P'$, and the edge connects two nodes $NV P(o_i)$ and $NV P(o_j)$ if there is an edge $(v_i, v_j) \in E_r$ connecting $v_i \in NV P(p_i)$ and $v_j \in NV P(p_j)$. Figures 5a to c show a NVD built for $P'$ bar $= \{p_1, p_2, p_3, p_4, p_5\}$. As shown in Figure 5b and c, $G_r$ is divided into five regions (NVPs). The NVPs of $\{p_1,p_2,p_3\}$ are mutually adjacent. Each NVP contains the vertices with the same NN. Using NVDs, the NN of any vertex $v \in G_r$ can be obtained directly. For example, the NN of a user located on $v_1$ is $p_2$, as $v_1 \in NV P(p_2)$.

Let Voc be the vocabulary of keywords in $P$. For each $t_i \in Voc$, we build NVDs for each $P'$, and associate the NN information with each vertex in $G_r$. Based on the properties of NVDs [25], $k$ closest relevant POIs can be obtained to derive tightened score bounds to facilitate pruning (to be presented in Section 6.2). With such components, a nonleaf node $N_l$ of IG-NVD is of the form $(G_r, P_{tr}, InvF_r, B_l, NN_l)$. Here, (i) $G_r$ is the subgraph indexed by $N_l$, (ii) $P_{tr}$ is a set of pointers pointing to the child nodes of $N_l$, (iii) $InvF_r$ is the inverted file of $N_l$ indexing keyword information of users and POIs in child nodes, (iv) $B_l$ is the set of border vertices of $N_l$, and (v) $NN_l$ is a table of $N_l$ associating each $b \in B_l$ with the corresponding NN in the NVDs. For instance, in Figure 4a and b, given a tree node $N_1$ with
the NN of \( v_{11} \) in \( NV \bar{D} \) is \( p_3 \), as \( v_{11} \in NV \bar{P}(p_3) \) (i.e., Figure 5b); the record \( NN[v_{11}]|\bar{\prime}p_3\prime| = p_3 \) is maintained in \( NN_i \). A leaf node \( N_j \) is in a form \( (G_j, InvF_j, B_j, V_j^r, NN_j) \), here \( V_j^r \) is the vertex set of \( G_j \), \( NN_j \) associates each vertex \( v \in V_j^r \) with NN information, and the remaining elements are similar as those in nonleaf nodes. Finally, instead of using G-Tree to compute shortest path distances (whose time complexity is \( O(|V_r|) \) [56]), IG-NVD also combines with an auxiliary component, the pruned highway label (PHL) [3], for more efficient network distance calculation with nearly constant time costs on massive road networks.

The IG-NVD index is fundamental in our framework, as the time complexity of batch queries processing based on this index is alleviated to \( O(\xi_1 \cdot |U| + \xi_2) \) \( (\xi_1 \ll \xi_2 \ll P) \), which is more scalable in practice (to be discussed in Section 6.3). In addition, the index size of IG-NVD (the major part of the space costs in the algorithm) is also scalable in practice, as two optimizations used as below.

**Optimizations** It is nontrivial to index all the NVDs and NN information in IG-NVD without excessive space costs, especially when the road network and the keyword vocabulary are large. To alleviate space costs and improve the scalability of the index, we further optimize IG-NVD from two aspects. i) The first optimization is to reduce the total number of NVDs by only constructing NVDs for frequent keywords while keeping the information of infrequent keyword in the inverted posting lists [1]. As most of keywords in real-world datasets occur infrequently according to the Zipfian distribution [33] \(^2\), we can void constructing NVDs for many infrequent keywords, thus significantly reduce the total index size. ii) The second optimization is to reduce the average size of each NVD by sharing redundant NN records. Since many frequent keywords are distributed unevenly in the road network, many vertices in \( G_r \) have the identical NN results. For such keywords, we only index compressed NN information by sharing redundant records and thus reduce the average space cost of each NVD. With such optimizations, the index size of IG-NVD can be substantially reduced. As shown in Figure 6, the index size of the IG-NVD index without the optimizations is about 1–2 orders of magnitude larger than the IG-NVD index with optimizations. In particular, the IG-NVD index without optimizations scales poorly in our largest dataset, Twitter (requires more than 80 GB space consumption), while the IG-NVD index with optimizations scales well over all datasets (the largest space consumption is

---

\(^2\) The Zipfian distribution indicates that most of keywords in the datasets appear infrequently [1]
This empirical result validates the effectiveness of above optimizations in improving the scalability of IG-NVD in practice.

6.2 Pruning schemes

Existing techniques [12, 28] derive separate bounds on spatial, textual, and social similarity scores [55] and then combine them to obtain overall estimated bounds. This simple bound aggregation may yield loose overall bounds, which incurs poor pruning abilities, especially for sparse social data. To address this, according to the definitions of BRNN, we group relevant users of each $p \in P$ into two categories: the set $\mathcal{U}_s(p)$ of users with social relevance to $p$ and the set $\mathcal{U}_t(p)$ of users with textual relevance. Different type of users are evaluated separately with tailored pruning rules.

6.2.1 Pruning socially relevant users

Observation 1 In real settings, the check-in data is sparse due to the limited mobility of users (i.e., users are more likely to visit the places in their local area). Hence, most POIs have relatively few check-ins, which means that the average number of socially relevant users for each POI is relatively small.
The sparsity of social data impedes bound estimation. Instead of combining social relevance bounds with other similarity bounds, we extract users with social relevance independently and estimate their top-k scores with only geo-textual context. Due to Observation 1, this is expected to efficient in practice (as shown in Section 8).

Lemma 4 Given a user $u_t$, the scoring lower bound list $\mathcal{L}_{SB}^{-1}(u_t)$ of $u_t$ is a list of $k$ tuples $(p_j, s_j)$ sorted in descending order of $s_j$. Here, $s_j$ is the geo-textual similarity score between $p_j$ and $u_t$, i.e., $s_j = (1 - \alpha) \cdot \frac{f(u_t, p_j)}{f_i(u_t, p_j)}$. For a POI $p$, if $F_{GST}(u_t, p) \geq s_k$, we have $u_t \notin S_i(p)$.

Proof Assume that $u_t \in S_i(p)$ (i.e., $u_t$ is in the BR$kNN$ result set of $p$). According to Definition 2, $p$ should be in the top-$k$ results of $u_t$. Hence, we have $s_k \leq F_{GST}(u_t, p)$, which contradicts the condition of the lemma. □

Algorithm 1 shows the procedures of computing $\mathcal{L}_{SB}^{-1}(u)$ with the IG-NVD index. The algorithm initializes $\mathcal{L}_{SB}^{-1}(u)$ and $\mathcal{PQ}$ as empty (line 1), where $\mathcal{PQ}$ is a priority queue to sort the keywords $t \in u.key$ in descending order of their geo-textual scores. For each $t \in u.key$, a separate max-priority $\mathcal{H}$ is maintained to store relevant POIs $p$ (lines 2–9). Specifically, if $t$ is a frequent keyword, the nearest POI $p_{NN}$ of $u$ in $\mathcal{P}$ is first assessed and inserted into $\mathcal{H}$ (lines 4–7). If $t$ is infrequent, all POIs in the inverted list of $t$ are assessed (line 9). When $|\mathcal{L}_{SB}^{-1}(u)| \leq k$, a while loop continues to assess more local relevant POIs by moving the top POI $p_n$ from $\mathcal{PQ}$ to $\mathcal{L}_{SB}^{-1}(u)$ (lines 12–15), and inserts each adjacent POI of $p_n$ in $\mathcal{NVD}_{\bar{u}_t}$ into $\mathcal{H}_{\bar{u}_t}$ for further evaluation (lines 16–19). Finally, $\mathcal{L}_{SB}^{-1}(u)$ is returned when it contains $k$ POIs (line 20). Next, we give a running example to illustrate how Algorithm 1 works.

Example 4 As shown in Figure 2, the socially relevant users of $p_1$ are $\{u_2, u_3\}$. Given $k = 2$, we first illustrate the computation of $\mathcal{L}_{SB}^{-1}(u_2)$. Since $u_2.key = \{\bar{bar}\}$, which is a frequent keyword, and $u_2$ is located within the network voronoi partition $N_{\bar{V}}(p_3)$ of $p_3$ in $N_{\bar{D}}$ (cf. Figure 5c), hence, the nearest POI to $u_2$ covering ‘bar’ is $p_2$, we insert $(p_2, 0.41)$ into $\mathcal{H}_{\bar{u}_2}$ (lines 12–15), and insert each POI of $p_2$ in $\mathcal{NVD}_{\bar{u}_2}$ into $\mathcal{H}_{\bar{u}_2}$ for further evaluation. In the first iteration, $(p_2, 0.41, 0.26)$, $\bar{bar}$ is inserted into $\mathcal{PQ}$ (lines 16–19). In the second iteration, $(p_1, 0.26, 0.17)$ is popped out from $\mathcal{PQ}$, and we move $(p_1, 0.26)$ from $\mathcal{H}_{\bar{u}_2}$ into $\mathcal{L}_{SB}^{-1}(u_2)$. The adjacent POIs of $p_1$ in $N_{\bar{D}}$ are $p_2$ and $p_3$. Since $p_2$ and $p_3$ have already been inserted in the former steps, neither of them are inserted into $\mathcal{H}_{\bar{u}_2}$. Now $\mathcal{H}_{\bar{u}_2} = \{(p_3, 0.17)\}$, and $(p_3, 0.17, 0.26)$ is added to $\mathcal{PQ}$. Finally, $|\mathcal{L}_{SB}^{-1}(u_2)| = 2$, the while-loop stops, and we get $\mathcal{L}_{SB}^{-1}(u_2) = \{(p_2, 0.41), (p_1, 0.26)\}$. Similarly, we obtain $\mathcal{L}_{SB}^{-1}(u_3) = \{(p_1, 0.33), (p_5, 0.18)\}$. Given $F_{GST}(u_2, p_1) = 0.26$ and $F_{GST}(u_8, p_1) = 0.08$, we can safely prune $u_8$ for $p_1$ by Lemma 4 as $F_{GST}(u_8, p_1) = 0.08 < \mathcal{L}_{SB}^{-1}(u_8) \cdot s_{u_8} = 0.18$, but $u_2$ cannot be pruned since $F_{GST}(u_2, p_1) = 0.26 \geq \mathcal{L}_{SB}^{-1}(u_2) \cdot s_{u_2} = 0.26$. Springe
6.2.2 Pruning textually relevant users

Let $\mathcal{U}_T(p)$ denote the set of users with textual relevance to a POI $p$. Intuitively, we can also utilize Algorithm 1 to prune the users in $\mathcal{U}_T(p)$. Nonetheless, the number of users with textual relevance to $p$ can be large, which means $|\mathcal{U}_T(p)| \ll |\mathcal{U}_T(p)|$. Therefore, if we evaluate each $u \in \mathcal{U}_T(p)$ one by one, it would cause unnecessary computation costs. In view of this, we develop pruning schemes tailored for textually relevant users with the notion of pseudo-users.

**Definition 6** Pseudo-user. Given a POI $p$, a G-Tree node $N_r$, and a border vertex $b_i \in \mathcal{B}_r$, a pseudo-user $u_{ps} = (\text{loc}, \text{Area}, \text{key})$ is a synthetic data point built on $b_i$, with $u_{ps}.\text{loc} = b_i$, $u_{ps}.\text{Area} = N_r$, and $u_{ps}.\text{key} = \mathcal{U}_i.\text{key}_u \cap p.\text{key}$. Here $\mathcal{U}_i$ is the set of users in $N_r$, and $\mathcal{U}_i.\text{key}_u$ is the union of the keywords covered by all users $u \in \mathcal{U}_i$.

Unlike pruning in Euclidean spaces [12], the notion of pseudo-users is proposed to prune road network spaces (which is more complex) by verifying a few landmark border vertices along shortest paths. By utilizing pseudo-users, tighter bounds can be derived that facilitate the pruning lemmas presented as below.
Lemma 5  Given a POI $p$, a pseudo-user $u_{ps} = (b_i, N_i, \mathcal{U}_i.key_u \cap \text{p.key})$, where $b_i$ is one of the border vertices of $N_i$ (i.e., $b_i \in \mathcal{B}_i$). Let $\mathcal{U}_i(p)$ be the users in $\mathcal{U}_i$, who are textually relevant to $p$ and whose shortest paths to $p$ pass through $b$. Then, for $\forall u \notin \mathcal{U}_i(p)$, we have $F_{GST}(u,p) \leq F_{GST}(u_{ps},p)$.

Proof  The shortest path from the POI $p$ to any $u \in \mathcal{U}_i(p)$ must pass the border vertex $b$. Thus, $f_i^p(u_{ps},p) \leq f_i^p(u,p)$ holds for each $u \in \mathcal{U}_i(p)$. Next, as $\mathcal{U}_i(p).key_u \subseteq \mathcal{U}_i.key_u$, $f_i(u_{ps},p) \geq f_i(u,p)$ holds for each $u \in \mathcal{U}_i(p)$. Combined with Eq. 1, for each user $u \in \mathcal{U}_i(p)$, we have $F_{GST}(u,p) \leq F_{GST}(u_{ps},p)$. The proof completes. □

Lemma 6  Given a pseudo-user $u_{ps} = (b,N_i, \mathcal{U}_i.key_u \cap \text{p.key})$, if $\mathcal{L}_i^{SB}(t).s_k \geq F_{GST}(u_{ps},p)$ holds for each $(t, \mathcal{L}_i^{SB}(t)) \in \mathcal{M}_i^{SB}(u_{ps})$, none of textually relevant users in $\mathcal{U}_i(p)$ can be influenced by $p$, and thus $u_{ps}$ can be discarded.

Proof  By Lemma 5, $F_{GST}(u_{ps},p) \geq F_{GST}(u,p)$ holds for any $u \in \mathcal{U}_i(p)$. If all $t \in u_{ps}.key$ having $\mathcal{L}_i^{SB}(t).s_k \geq F_{GST}(u_{ps},p)$, it must hold that $\mathcal{L}_i^{SB}(t).s_k \geq F_{GST}(u,p)$ for all users in $\mathcal{U}_i(p)$. Hence, there are at least $k$ POIs $p_i(1 \leq i \leq k)$ in $\mathcal{L}_i^{SB}(t)$ with $F_{GST}(u,p_i) \geq s_k \geq F_{GST}(u_{ps},p) \geq F_{GST}(u,p)$. Hence, no user $u \in \mathcal{U}_i(p)$ can be influenced by $p$, and $u_{ps}$ can be pruned. The proof completes.

Lemma 7  Given a tree node $N_i$, if all pseudo-users $u_{ps}$ generated for $p$ on each border $b_i \in N_i$ satisfy the condition in Lemma 6, all the textually relevant users with shortest paths to $p$ passing any vertex in $N_i$ cannot be influenced by $p$.

Proof  Lemma 7 is an extension of Lemma 6.

The procedure of computing $\mathcal{L}_i^{SB}(t)$ for each frequent keyword $t \in u_{ps}.key$ when generating $\mathcal{M}_i^{SB}(u_{ps})$ resembles that in Algorithm 1, and hence, we omit it for brevity.

6.3 Batch processing algorithm

By exploiting our proposed index and the pruning rules mentioned above, the pseudocode of the batch BR$k$NN processing algorithm is presented in Algorithm 2. Two important functions of Algorithm 2 are shown in Algorithm 3. Algorithm 2 initializes a priority queue $Q$ to maintain pairs of each generated pseudo-user and the corresponding POI in ascending order of the shortest path distances. A set $\mathcal{U}_c$ stores all BR$k$NN result candidates, a vector $\mathcal{N}_{max}$ maintains the current uppermost nodes explored for each POIs, and a map $\mathcal{A}^{\mathcal{L}}$ associates each user $u \in \mathcal{U}_c$ with the POIs in $\mathcal{P}_c$ that may influence $u$ (line 1). Let $\text{Leaf}(p)$ denote the leaf node where a POI $p$ is located, and let $\hat{\mathcal{U}}_p(p, \text{Leaf}(p))$ be the set of users located in $\text{Leaf}(p)$ with textual relevance to $p$ (i.e., the local textually relevant users). Algorithm 2 first evaluates socially relevant users as well as local textually relevant users for each $p \in \mathcal{P}_c$ (lines 2–7). Then, it iteratively expands the road network, and evaluates the remaining textually relevant users to find more BR$k$NN candidates (lines 8–13). During
this process, pseudo-users are generated on the border vertices at different levels of the G-Tree (lines 6–10 of Algorithm 3) to effectively prune users by Lemmas 6 and 7 (lines 12 and 17 of Algorithm 3). Users who cannot be pruned are added to $U_c$ (line 6) and further verified by the top-$k$ algorithm extended from an existing algorithm [1] (lines 15). Note that we modify the existing top-$k$ algorithm [1] to incorporate social relevance. Finally, batch BR$\kappa$NN results for each $p \in P_c$ are returned (line 18). To better illustrate the procedures in Algorithm 2, we give an example as below.

**Example 5** Given $P_c = \{p_1, p_3\}$ and $k = 2$. Algorithm 2 first evaluates socially and local textually relevant users (lines 2–7). Since each step is similar to Example 4, we omit its details for brevity. After that, we obtain $U_c = \{u_2, u_3, u_4, u_5\}$, $\mathbb{N}_{\text{max}} = \{p_1, N_4\}$, $\langle p_3, N_4 \rangle$, and $\mathcal{A} = \{(u_2, \{p_1\}), (u_3, \{p_1, p_3\}), (u_4, \{p_3\}), (u_5, \{p_3\})\}$. The rest of textually relevant users are evaluated (lines 8–13) as follows.

**Loop 1:** As $Q$ is empty, $\mathbb{N}_{\text{max}} = \{p_1, N_4\}$, $\langle p_3, N_4 \rangle$, the Extend_Range function is invoked to explore the G-tree (line 9). In Figure 4, the nearest ancestor node of $N_4$ with textual relevance to $p_1$ and $p_3$ is $N_1$ (line 4–5 of Algorithm 3). And the child nodes of $N_1$, (which is also the sibling node of $N_4$) with textual relevance to $p_1$ and $p_3$ is $N_3$. Hence, two pseudo-users $u_{ps1} = (v_9, N_3, \{'bar', 'wine'\})$ and $u_{ps2} = (v_9, N_3, \{'bar'\})$ are generated on the border vertex $v_9$ of $N_3$ for $p_1$ and $p_3$, respectively, with $(u_{ps1}, p_1)$ and $(u_{ps2}, p_3)$ pushed
into $Q$ (line 6–10 of Algorithm 3). After this, $Q = \{((u_{p2},p_3),5),((u_{p1},p_1),10)\}$. Now, $N_1$ becomes the current uppermost node explored for $p_1$ and $p_3$. As $N_1$ can be pruned for $p_1$ but cannot be pruned for $p_3$ via Lemma 7, the tuple $\{p_1,N_1\}$ is removed from $\mathbb{N}_{\text{max}}$, with $\{p_3,N_4\}$ in $\mathbb{N}_{\text{max}}$ replaced by $\{p_3,N_1\}$, and we update $\mathbb{N}_{\text{max}} = \{p_3,N_1\}$ (lines 11–13 of Algorithm 3).

Next, we proceed to pop out $(u_{p2},p_3)$ from $Q$ (line 10), where $u_{p2} = (v_9,N_3,\{\text{‘bar’}\})$. $\mathcal{M}_{\text{SB}}^1(u_{p2})$ is generated by computing $\mathcal{L}_{\text{SB}}^1(\text{‘bar’})$ (line 12). As $N_1$ is a leaf node and the keyword of $u_{p2}$ is ‘bar’, which can not be pruned by lemma 6 (lines 15–18 of Algorithm 3), the textually relevant users $u_1$ and $u_2$ in $N_3$ are evaluated for $p_3$ (line 21 of Algorithm 3). As $\mathcal{L}_{\text{SB}}^1(u_1).s_2 \geq F_{\text{GST}}(u_1,p_3)$ and $\mathcal{L}_{\text{SB}}^1(u_2).s_2 \geq F_{\text{GST}}(u_2,p_3)$, both $u_1$ and $u_2$ can be pruned for $p_3$ by Lemma 4 (lines 5–6).

**Loop 2:** As $Q$ is not empty, we pop out $(u_{p1},p_1)$ from $Q$, where $u_{p1} = (v_9,N_3,\{\text{‘bar’},\text{‘wine’}\})$. Similarly, $u_{p1}$ cannot be pruned, and thus textually relevant users of $p_1$ in $N_3$ (i.e., $u_1$ and $u_2$) are evaluated. As $\mathcal{L}_{\text{SB}}^1(u_1).s_2 \leq F_{\text{GST}}(u_1,p_3)$ and $\mathcal{L}_{\text{SB}}^1(u_2).s_2 \leq F_{\text{GST}}(u_2,p_3)$, $u_1$ and $u_3$ are added to $\mathcal{U}_c$ for further verification.

Again, $Q$ becomes empty, but as $\mathbb{N}_{\text{max}} \neq \emptyset$, we continue to explore the road network, update $Q$, and examine the remaining entries until $Q = \emptyset$ and $\mathbb{N}_{\text{max}} = \emptyset$. After the while loop stops, we get $\mathcal{U}_c = \{u_1,u_2,u_3,u_4,u_5\}$, which is further examined by the top-$k$ algorithm. Details of the procedures in top-$k$ verification can be found elsewhere [1]. Finally, we get batch BR$k$ NN results as $S_c(p_1) = \{u_1,u_2,u_3\}$ and $S_c(p_3) = \{u_3,u_4,u_5\}$.

**Discussion** We analyze the time complexity of Algorithm 2, which includes three major parts: i) evaluating socially and nearby textually relevant users (lines 2–7), ii) evaluating remaining textually relevant users in BR$k$NN candidates (lines 14–17). In the worst case, Algorithm 2 accesses at most $|\mathcal{U}|$ users for each POI during part i) and ii), the time complexity of which is $O(|\mathcal{U}| |\mathcal{P}| \delta_1)$, where $\delta_1$ denotes the time complexity of Algorithm 1. Let $\tau$ be the largest frequency of an infrequent keyword, and let $\bar{w}_1$ ($w_2$) be the average number of frequent (infrequent) keywords of a user. By analyzing Algorithm 1, we have $O(\delta_1) = O(k(\bar{w}_1 + \bar{w}_2) + (\tau + \log \tau)\bar{w}_2 + k\log k)$. Further, at most $|\mathcal{S}|$ (the number of border vertices) pseudo-users are generated for each POI, the complexity of which is $O(|\mathcal{S}| |\mathcal{P}| \delta_2)$. Here, $\delta_2$ represents the cost of computing $\mathcal{M}_{\text{SB}}^1(u_{p1})$ for a pseudo-user $u_{p1}$, which is $O(k\bar{w}_1 + \tau\bar{w}_2 + k\log k + \tau \log \tau)$, with $\bar{W}_1$ ($\bar{W}_2$) being the average number of frequent (infrequent) keywords covered by a POI. Finally, let $\Delta$ be the cost of the top-$k$ algorithm. The cost of phase iii) is $O(|\mathcal{U}| |\mathcal{P}| \Delta)$. To sum up, the total time complexity of Algorithm 2 is $O(\xi_1|\mathcal{U}| + \xi_2)$, with $\xi_1 = (\Delta + \delta_1)|\mathcal{P}|$ and $\xi_2 = \delta_2|\mathcal{P}| |\mathcal{S}|$. As $\tau$, $\bar{w}_1$, $\bar{w}_2$, $|\mathcal{S}|$, and $k$ are often small values, while $|\mathcal{S}|$ and $\Delta$ are also small compared with $|\mathcal{U}|$ and $|\mathcal{P}|$ [1, 47], we conclude that $\xi_1$ and $\xi_2$ are reasonably small values (i.e., $\xi_1(\xi_2) \ll |\mathcal{P}|$).

### 7 POI selection policies

Next, we present methods to effectively evaluate the influence of users in BR$k$NNs, then we give alternative policies to answer MaxInfBR$k$NN with qualified results.
7.1 Effective influence evaluation

Given a POI set \( \mathcal{P}_s \), and an RR set collection \( \mathcal{K} \), instead of only using the coverage of RR set to derive the influence estimation \( \mathbb{I}_p^+(\mathcal{P}_s) \) (i.e., which is inaccurate when \( \mathbb{I}_p(\mathcal{P}_s) \) is small), we evaluate the influence in a hybrid manner:

\[
\mathbb{I}_p(\mathcal{P}_s) = \mathbb{I}_h^+(\mathcal{P}_s) + \mathbb{I}_h^-(\mathcal{P}_s) + A_{\mathcal{R}}(\mathcal{P}_s) \cdot |\mathcal{V}_s| / |\mathcal{R}|
\] (4)

In Eq. 4, \( \mathbb{I}_h^+(\mathcal{P}_s) \) is the local influence of \( \mathcal{P}_s \), which is propagated from users in \( \mathcal{S}_s(\mathcal{P}_s) \) at most \( h \)-hops. \( \mathbb{I}_h^-(\mathcal{P}_s) \) is the remote influence estimation of \( \mathcal{P}_s \), which approximates the number of users influenced by \( \mathcal{S}_s(\mathcal{P}_s) \) further than \( h \)-hops away, and is measured by the number (i.e., \( A_{\mathcal{R}}(\mathcal{P}_s) \)) of covered RR sets. Using this formulation, though the exact social influence computation is hard to capture, the exact \( h \)-hop based local influence can be accurately and efficiently computed [44, 45]. As computing exact local influence is expensive when \( h \geq 3 \), we fix \( h = 2 \) as a trade-off between efficiency and accuracy [44, 45]. To estimate the remote influence with RR sets, it worth mentioning that the RR set should be modified to capture the influence beyond 2-hops, so, any POI \( p \) is removed from the original RR set when users in \( \mathcal{S}_s(p) \) are reached during the sampling within 2 hops.

7.2 Approximation solution

Revisiting the bounds in Lemma 3 We first greedily select \( b \) POIs to maximize the objective function in Eq. 4, and denote this POI set as \( \mathcal{P}_b^{H^*} \). We also extend an existing 2-hops based greedy algorithm [44] to generate a POI set with approximately optimal local influence. Here, the 2-hops based greedy algorithm [44] is a modified greedy hill-climbing algorithm. It starts with an empty POI set, and greedily selects the POI whose BR has the largest marginal gain to the influence within two hops (i.e., the marginal local influence). The algorithm terminates when it selects \( b \) POIs. We denote the POI set selected by the 2-hops based greedy algorithm as \( \mathcal{P}_b \). Let \( \mathcal{P}_b(\mathcal{P}_c) \) be the collection of all size-\( b \) subsets of \( \mathcal{P}_c \) (i.e., \( \mathcal{P}_b = \{ \mathcal{P}_b | \mathcal{P}_b \subseteq \mathcal{P}_c \land |\mathcal{P}_b| = b \} \)), and let \( \mathcal{P}_b^{L^*} (\mathcal{P}_b^{H^*}) \) be the POI set with the largest influence (local influence) among all elements in \( \mathcal{P}_b^* \). Next, we revise the influence bounds as follows.

i) Derivation of \( \mathbb{I}_p^-(\mathcal{P}_b^{H^*}) \). Combining (2) of Lemma 3 with (4), we get the following revised lower bound:

\[
\mathbb{I}_p^-(\mathcal{P}_b^{H^*}) = \mathbb{I}_h^+(\mathcal{P}_b^{H^*}) + \left( \sqrt{A_{\mathcal{R}_2}(\mathcal{P}_b^{H^*}) + \frac{2\eta_1}{9} - \frac{\eta_1}{2}} \right)^2 \cdot |\mathcal{V}_s| / |\mathcal{R}_2|.
\] (5)

ii) Derivation of \( \mathbb{I}_p^+(\mathcal{P}_b^{H^*}) \). As submodularity also holds for the local influence computation [45], we have \( \mathbb{I}_h^+(\mathcal{P}_b^{H^*}) = \mathbb{I}_h^+(\mathcal{P}_b^{L^*}) + \mathbb{I}_h^+(\mathcal{P}_b^{H^*})/(1 - 1/e) \). By combining this with (4) and (3), we derive the following upper bound:

\[
\mathbb{I}_p^+(\mathcal{P}_b^{H^*}) = \mathbb{I}_h^+(\mathcal{P}_b^{L^*})/(1 - 1/e) + \left( \sqrt{A_{\mathcal{R}_1}(\mathcal{P}_b^{H^*}) + \frac{\eta_1}{2} + \sqrt{\frac{\eta_1}{2}}} \right)^2 \cdot |\mathcal{V}_s| / |\mathcal{R}_1|.
\] (6)

The maximum number of RR sets ensuring the approximation ratio is formulated in Lemma 8, revised from [43].
Lemma 8 [43] Given a set \( \mathcal{R} \) of RR sets and parameters \( \epsilon \) and \( \delta \), if \(|\mathcal{R}|\) is larger than \( \theta_{\text{max}} \):

\[
|\mathcal{R}| \geq \theta_{\text{max}} = \frac{2|\mathcal{V}_s| \cdot \left( (1 - 1/e) \sqrt{\ln \frac{6}{\delta}} + \sqrt{(1 - 1/e) \left( \ln \left( \frac{|\mathcal{P}_c|}{b} \right) + \ln \frac{6}{\delta} \right)} \right)^2}{\epsilon^2 \mathbb{I}_p(\mathcal{P}_b^{\text{opt}})} ,
\]

then the approximation ratio of \( \mathcal{P}_b^{\text{H*}} \) is no worse than \( 1 - 1/e - \epsilon \) with a probability of at least \( 1 - \delta \).

To approximate the value of \( \mathbb{I}_p(\mathcal{P}_b^{\text{opt}}) \) in Lemma 8, an efficient and tightened lower bound computation can be \( \mathbb{I}_p(\mathcal{P}_b^{\text{L*}}) \) (denoted as \( \Psi \)). According to \( \theta_{\text{max}} \), we can first sample \( \theta_0 \) RR sets to initiate the RIS process, where \( \theta_0 \) is computed as follows.

\[
\theta_0 = \theta_{\text{max}} \cdot \epsilon^2 \cdot \Psi / |\mathcal{V}_s| \quad (7)
\]

With the above formulations, the approximation solution (denoted as AP) to answer MaxInfBR^NN is presented in Algorithm 4. AP first invokes Algorithm 2 to retrieve batch results for each \( p \in \mathcal{P}_c \) and then constructs the heterogeneous graph \( \mathcal{G}_H \) (lines 1–2). Next, it obtains \( \mathcal{P}_b^{\text{L*}} \) to derive a tight lower bound on \( \mathbb{I}_p(\mathcal{P}_b^{\text{opt}}) \), and then it initializes \( \rho_{\text{max}} \) and \( \theta_0 \) by Lemma 8 and (7), respectively (lines 3–5). After this, AP samples two RR set collections \( \mathcal{R}_1 \) and \( \mathcal{R}_{2\rho} \) with \( |\mathcal{R}_1| = |\mathcal{R}_2| = \theta_0 \) to initiate the sampling phase with at most \( i_{\max} = \left\lceil \log_2 \frac{\theta_{\text{max}}}{\theta_0} \right\rceil \) iterations (lines 8–15). In each iteration, AP utilizes \( \mathcal{R}_1 \) to obtain \( \mathcal{P}_b^{\text{H*}} \).
which approximately maximizes the objective function in (4) (line 9). Next, it utilizes $\mathcal{R}_2$ to derive $\mathbb{I}_p^-(P^H_b)$ by (5) (line 10). It further computes $\mathbb{I}_p^+(P^H_{br})$ using $\mathbb{P}^c_{br}$ and $\mathcal{R}_1$ by (6) (line 11) and sets the approximation ratio to $\mathbb{I}_p^-(P^H_b) / \mathbb{I}_p^+(P^H_{br})$ (line 12). AP terminates and returns $P^H_b$ if a $(1 - 1/e - \epsilon)$ approximation ratio is achieved (lines 13–14). Otherwise, it doubles the RR sets in $\mathcal{R}_1$ and $\mathcal{R}_2$ and perform a new iteration (line 15).

**Discussion** Algorithm 4 initializes parameters $\eta_l$ and $\eta_u$ in (5) and (6) as $\ln(3i_{\max}/\delta)$ (line 7), so parameters $\delta_l$ and $\delta_u$ in Lemma 3 are fixed at $\delta/3i_{\max}$. In this setting, the probability of deriving an incorrect approximation ratio in each iteration is at most $2\delta/3i_{\max}$. By the union bound for the first $(i_{\max} - 1)$ iterations, the probability that Algorithm 4 fails to find a qualified result is less than $2/3\delta$. Also, in the last iteration, by Lemma 8, Algorithm 4 ensures that $P^H_b$ is a $(1 - 1/e - \epsilon)$ approximation result with a probability of $(1 - \delta/3)$. Overall, AP guarantees a $(1 - 1/e - \epsilon)$ approximate solution with probability at least $1 - \delta$.

### 7.3 Heuristic solution

As an alternative solution to MaxInfBRkNN, we proceed to develop a heuristic approach (denoted as HE) by alleviate some costly procedures in AP. HE has no rigorous approximation ratios, it trades accuracy for efficiency in practice.

**Aggressive pruning and compensation** When a POI in $P$ contains less frequent keywords, the conditions in line 18 of Algorithm 3 may be too strict to prune any pseudo-users. To enable pruning, instead of computing $S_{GB}^t(i)$ for $i \in u_{ps.key}$, we only consider the keywords that occur frequently in users. In other words, keywords rarely appeared in users are pruned aggressively. To ensure the correctness of batch results, users with infrequent keywords are evaluated separately by Algorithm 1 for compensation. This improves the pruning efficiency of Algorithm 2.

**Assign priorities to BRkNN candidates** Since our problem focuses on the most influential POIs, it is unnecessary to exactly retrieve BRkNNs for each $p \in P_c$. Hence, we revise lines 14–17 of Algorithm 2 as follows. Let $\hat{S}_h(p)$ ($\hat{S}_c(p)$) be the set of candidate users (potential users) currently obtained (verified) for $p \in P_c$. TkGSKQ verification is performed for each $u \in U_c$ in descending order of user priority. The priority of a user $u$ is given as $|F(u)| \cdot |A_{\mathcal{L}}(u)|$, since a candidate user $u$ associated with more friends and POIs is likely to be more influential. By this, with more influential users being verified earlier in lines 14–17 of Algorithm 2, the difference between $\hat{S}_h(p)$ and $\hat{S}_c(p)$ decreases, and it is possible to find a highly qualified result without verifying all $u \in U_c$ based on the strategies covered next.

**POI selection by approximating local influence** About 90% of the influence propagation occurs within 3 hops under the IC model [44, 45]. Hence, the influence within 2-hops can used to determine the quality of POI sets. Let $G_H$ and $G_H$ be the heterogeneous graph built on $\hat{S}_h(p)$ and $\hat{S}_c(p)$ for each $p \in P_c$. Then, each time when a small batch of candidate users is verified, we perform the 2-hops based greedy algorithm on $G_H$ and $G_H$ to get sets $P^L_b$ and $P^C_b$, respectively. If the accuracy satisfies $\mathbb{I}_p^L(P^L_b) / \mathbb{I}_p^C(P^C_b) \geq 1 - 1/e$, we return $P^L_b$ as the result. $P^L_b$ is expected to have good quality, due to the effectiveness of 2-hop based influence estimation [45].

© Springer
8 Performance study

This section experimentally evaluates the proposed methods. All the methods are implemented in C++, tested on a PC with an Intel i7 2.90GHz CPU and 64GB RAM.

8.1 Experimental settings

Datasets We employ three datasets: LasVegas, Gowalla, and Twitter. In particular, LasVegas is a real-world dataset obtained from the Yelp3, which contains POIs and users located in the city of Las Vegas. Each POI contains an ID, a textual description, and latitude and longitude. Each user contains check-ins, reviews of POIs, and IDs of friends. The road network of LasVegas is obtained from OpenStreetMap4, and we map each POI and user to the closest vertex on the road network. Gowalla and Twitter are synthetic datasets generated by combining road networks 5 with geo-tagged social networks extracted from existing datasets of [27]. The keywords in Gowalla and Twitter are created to follow Zipfian distributions [33]6. Following the IM literature [24, 27, 43], we utilize the weighted cascade model [27, 46] to set the weight of each edge \((u, v)\) in \(G_s\) to \(1/d_{in}(v)\), where \(d_{in}(v)\) is the in-degree of user \(v\) in \(G_s\). The statistics of all the datasets are given in Table 3.

Query set generation We obtain the POIs in the query sets as follows. First, we randomly select two frequent keywords (such as 'bar' and 'shopping mall') from the dataset vocabulary. Then, for each keyword, we extract the group of POIs 7 covering this keywords and eliminate those without any check-ins. Each group of POIs is further divided into two parts based on the popularity of POIs among users (whether the number of check-ins exceeds the average check-ins per POI). Finally, we have four clusters of POIs \((P_{c1} \text{ through } P_{c4})\), and

| Table 3 Statistics of the datasets          | LasVegas | Gowalla | Twitter |
|-------------------------------------------|---------|---------|---------|
| # of road network vertices               | 43,401  | 1,070,376| 1,890,815|
| # of road network edges                  | 109,160 | 2,712,798| 4,630,444|
| # of social network vertices             | 87,990  | 196,228 | 2,062,280|
| # of social network edges                | 552,486 | 1,900,654| 4,803,440|
| # of POIs                                | 28,851  | 218,659 | 326,662 |
| # of users (with reviews)                | 28,867  | 196,228 | 1,114,261|
| average # of keywords per POI            | 350.2   | 8.1     | 4.4     |
| average # of keywords per user           | 50.2    | 3.1     | 2.8     |
| average # of check-ins per POI           | 63.3    | 5.0     | 5.3     |
| average # of friends per user            | 6.3     | 9.7     | 8.7     |

---

3 http://www.yelp.com/dataset_challange/
4 https://www.openstreetmap.org/
5 http://www.dis.uniroma1.it/challenge9/
6 Zipf’s law predicts that the frequency \(f_r\) of a keyword \(r\) in real-world datasets satisfies \(f_r \propto 1/r^\alpha\), where \(r\) is the rank of \(r\) and \(\alpha \approx 1\), which means most keywords in the datasets occur infrequently [1].
7 As POIs often have multiple keywords, infrequent keywords also can be covered by these POIs.
we randomly select \( |\mathcal{P}_c| \) POIs from each cluster into \( \mathcal{P}_c \), with \( \mathcal{P}_{c1} \) (i.e., the most popular POI set) as the default query set.

**Competitors** We first compare the performance of our index (i.e., IG-NVD) with the existing state-of-the-art index (i.e., the GIM-Tree) \[55\]. Then, to investigate the effectiveness of our problem in finding highly influential POIs, we compare the POI sets retrieved by Max-InfBR\(k\)NN queries with those retrieved by several simple but smart POI selection policies:

- **Relevance.** This policy greedily selects \( b \) POIs from \( \mathcal{P}_c \) with textual or social relevance to most of the users located in the local area (i.e., within 5\( km \) away).
- **Influencer.** This policy selects POIs whose \( BRk \) NNs include the largest number of influencers. Specifically, it disregards users without textual or social relevance to \( \mathcal{P}_c \), and utilizes existing IM algorithms \[43\] to find top-\( x \) (i.e., \( x = 200 \)) users as influencers. Then, it computes the top-\( k \) results for each influencer, and selects \( b \) POIs which are top-\( k \) relevant to the largest number of influencers.
- **MaxBR\(k\)NN.** This policy maximizes the cardinality of BR\(k\)NNs for any size-\( b \) POI set. We revise an existing algorithm \[12\] by considering road network distance and social relevance in the similarity function, and greedily select \( b \) POIs having the largest size of BR\(k\)NN results.
- **Random.** As a simplest competitor, this policy randomly selects \( b \) POIs from \( \mathcal{P}_c \).

**Parameter Settings** The parameter settings are listed in Table 4, with default values in bold. Note that in Table 4, \( |\mathcal{U}| \) (\( |\mathcal{P}| \)) denotes the percentage of all users (POIs) w.r.t. the number of users (POIs) for testing. In BA and AP, a trade-off between accuracy and efficiency is controlled by parameters \( \varepsilon \) and \( \delta \), respectively. Following settings extensively used in recent studies \[17, 43\], we set \( \varepsilon = 0.1 \) and \( \delta = 1/|\mathcal{V}_x| \), respectively.

| Parameter | Range |
|-----------|-------|
| \( k \) (\# of top-\( k \) ranking results) | 10, 15, 20, 25, 30 |
| \( |\mathcal{P}_c| \) (\# of candidate POIs) | 20, 40, 60, 80, 100 |
| \( b \) (the size of selected POIs) | 1, 3, 5, 7, 9 |
| \( \alpha \) (parameter in the ranking function) | 0.0, 0.2, 0.4, 0.6, 0.8 |
| \( |\mathcal{U}| \) (the cardinality of users) | 0.6, 0.7, 0.8, 0.9, 1.0 |
| \( |\mathcal{P}| \) (the cardinality of POIs) | 0.6, 0.7, 0.8, 0.9, 1.0 |

Values in bold are default parameter settings

| Dataset | GIM-Tree [55] | IG-NVD |
|---------|--------------|--------|
| Size (MB) | Time (sec) | Size (MB) | Time (sec) |
| LasVegas | 112 | 63 | 60 | 56 |
| Gowalla | 4,533 | 4,390 | 2,633 | 3,460 |
| Twitter | 42,438 | 49,633 | 3,913 | 5,868 |
8.2 Indexing performance

We compare the performance of our index (i.e., IG-NVD) with the existing index (i.e., the GIM-Tree [55]) for retrieving BR\(k\)NNs on geo-social networks. The statistics in Table 5 illustrate that, our index needs much less storage space than the GIM-Tree when the dataset becomes large. Specifically, the index size of the GIM-Tree is nearly 5 GB on Gowalla and increases to more than 40 GB on the largest dataset, Twitter, which may not fit in main memory on a commodity computer. In contrast, our index is only about 4 GB on Twitter, with a construction time of less than 2 hours. The two indexes have similar construction time on the smallest dataset (i.e., LasVegas). This is because there are few users and POIs in LasVegas, thus making the pre-processing costs on social information indexed in the GIM-Tree relatively low.

8.3 Effectiveness of POI selection

We report the influence of \(P\) returned by the different POI selection policies using the real dataset LasVegas. As shown in Table 6, for different kinds of POI candidate sets in LasVegas, MaxInfBR\(k\)NN consistently outperforms the competitors. Also, Influencer performs the best among the competitors, which indicates the importance of taking into account social influencers. However, the gap between Influencer and MaxInfBR\(k\)NN is large, because Influencer only selects POIs affecting top influencers (top-x influential users), while ignoring users with relatively less influence, which leads to considerable influence loss. In contrast, MaxInfBR\(k\)NN detects all the users in BR\(k\)NNs and comprehensively evaluates their influence; thus, MaxInfBR\(k\)NN can find highly qualified results.

8.4 Efficiency and scalability evaluation

We further utilize the larger synthetic datasets (i.e., Gowalla and Twitter) to comprehensively evaluate the scalability and efficiency of our proposed methods (i.e., BA, AP, and HE presented in Sections 3, 7.2, and 7.3, respectively). In each set of experiment, we vary a single parameter while fixing the remaining ones at their defaults.

We evaluate the impact of different parameter settings on our methods, in terms of i) the total runtime, ii) the influence of retrieved POI set (i.e., \(I_p(P_s)\)), iii) the BR\(k\)NN retrieval time, iv) the overhead incurred in POI selection and qualification (i.e., POI selection time and the number of sampled RR sets), and v) the approximation ratio of our methods against the exact solution. Recall that the total runtime of BA and AP consists of the BR\(k\)NN retrieval time and the POI selection time. The numbers at the top of the columns

| \(P_{c1}\) | \(P_{c2}\) | \(P_{c3}\) | \(P_{c4}\) |
|----------------|----------------|----------------|----------------|
| Relevance | Influencer | MaxBR\(k\)NN | Random | MaxInfBR\(k\)NN |
| 118 | 201 | 147 | 60 | 255 |
| 117 | 238 | 191 | 63 | 270 |
| 212 | 385 | 352 | 101 | 450 |
| 204 | 296 | 277 | 72 | 447 |

Values in bold are the largest influence values achieved by our compared policies.
in Figs. 7a–12a denote the percentage of the POI selection time in the total runtime. We do not report the number of sampled RR sets for HE, since HE utilizes heuristics instead of RR sets to select POIs.

Effect of $b$. The experimental results of our methods when varying $b$ are shown in Figure 7. We observe that, although BA, AP, and HE achieve similar influence (cf. Figure 7b), the running overhead of BA is significantly larger than those of AP and HE. Also, the performance of BA is not robust to variations in $b$. When $b = 1$, the total runtime of BA increases sharply (cf. Figure 7a), and nearly a billion RR sets are sampled and maintained by BA (cf. Figure 7d). The reason is that BA only utilizes the coverage of RR sets to estimate the influence. Thus, when $b$ is small (i.e., $b = 1$), relative estimation errors in BA grows, which forces BA to sample more RR sets to ensure the accuracy of the results. In contrast, AP and HE are more efficient and robust across different values of $b$. The BR$k$NN retrieval in AP is consistently 2 orders of magnitude faster than that in BA (cf. Figure 7c). The runtime of BR$k$NN retrieval in both BA and AP are unaffected by $b$ since BR$k$NN computation does not change much when $|\mathcal{P}_c|$ is fixed. The influence of each method increases with $b$ while the approximation ratio drops (cf. Figure 7e). This occurs because more POIs are included in $|\mathcal{P}_s|$, which magnifies the errors in selecting POIs greedily.

Note that in Figure 7e, we do not report the results of approximation ratios when $b = 9$. This is because to derive the approximation ratios, we must first obtain the exact solution by evaluating all the possible size-$b$ POI sets, which may cause unbearable running costs when $b$ becomes large. Recall that the default value of $|\mathcal{P}_c|$ is 60. Hence, the exact solution has to evaluate $\binom{60}{9}$ ($\approx 1.47 \times 10^{10}$) POI sets when $b = 9$, and we find that this process cannot terminate after 7 days (1 week). As Figure 7e aims to evaluate the accuracy of our methods when varying $b$. Hence, according to the approximation ratios by varying $b$ among $\{1,3,5,7\}$ as shown in Figure 7e, it is possible to conclude that approximation ratios of our methods drop with $b$.

Effect of $|\mathcal{P}_s|$. Figure 8 depicts the results when changing $|\mathcal{P}_s|$. The total runtime and the influence of each method increase with $|\mathcal{P}_s|$ (cf. Figure 8a and b), since more relevant users are evaluated during the process, and many of them become BR$k$NNs. Note that the runtime of BR$k$NN retrieval in BA is prohibitively large on the Twitter dataset (more than 10 hours),
and increases sharply with the growth of $|\mathcal{P}_c|$ (cf. Figure 8c). In comparison, BR$k$NN retrieval in AP runs much faster (only several minutes) and scales well as $|\mathcal{P}_c|$ grows. This illustrates the deficiency of existing methods [55] and confirms that our batch processing algorithms are more scalable on larger datasets. As indicated by the percentages on the top of the columns in Figure 8a, the POI selection time in both BA and AP accounts for less than 5% in most cases, implying that BR$k$NN retrieval is the query processing bottleneck. In Figure 8d, again, we find that the performance of BA is not robust and is sensitive to $|\mathcal{P}_c|$, while AP is relatively
robust. The approximation ratio of our methods decreases slightly with $|P_c|$, as more POIs are evaluated which increases the POI selection errors.

**Effect of $k$.** Figure 9 shows the results when varying $k$. As expected, AP and HE consistently outperform BA in terms of runtime, and all the methods achieve comparable influence. In particular, with the growth of $k$, the influence of the returned POI set as well as their approximation ratios increase for each method (cf. Figure 9b and e). This is because when $k$ becomes larger, more users can have some POIs in $P_c$ as their top-$k$ relevant
objects. Hence, the total size of potential users (influence set) for POIs in $P_c$ increases, which increases the influence and accuracy of $P_s$.

**Effect of $\alpha$.** Figure 10 illustrates the results when varying $\alpha$. Recall that $\alpha$ controls the balance between textual similarity and social relevance in the scoring function, and a higher value of $\alpha$ implies a higher preference for the social relevance. As expected, AP and HE consistently outperform BA in terms of running efficiency across different values of $\alpha$. The influence and the approximation ratio of each method are affected only little by $\alpha$, because the largest influence achieved by any size-$b$ POI set only changes little when changing $\alpha$.

**Effect of $|\mathcal{U}|$.** Figure 11 depicts the results when varying the cardinality of users in $\mathcal{U}$. It is seen that the runtime of BA increases with $|\mathcal{U}|$, while the runtime of AP and HE are relatively stable. This observation offers evidence of the effectiveness of our pruning techniques in pruning relevant users, and validate the superiority of AP and HE in tackling large datasets. The influence and the approximation ratio of each method increase with $|\mathcal{U}|$, as more users are influenced by $P_s$, which enhances the accuracy of our algorithms in selecting influential POIs. It is interesting that the POI selection time decreases with $|\mathcal{U}|$ (cf. Figure 11d). This is because as $|\mathcal{U}|$ increases, the influence of $P_s$ (i.e., $I_p(P_s)$) also increases, which decreases the relative errors between the approximate solution and the exact optimal solution. Hence all of our methods can find a qualified result earlier during the phase of POI selection.

**Effect of $|P|$.** Figure 12 plots the results when varying the cardinality of the POIs in $P$. As $|P|$ grows, the running time of each method drops. This is because the top-$k$ score of each user increases with the growth of $P$, enhancing the pruning power in all the methods. In addition, the influence and the approximation ratio of our methods drop as $|P|$ grows. The reason is that when $|P|$ is large, more competition exists in the top-$k$ ranking among POIs. Therefore, on average, fewer users can become the BR$\mathcal{R}$NN of each POI, yielding results with less influence and lower accuracy.
9 Conclusions

We identify a new geo-social networking problem, called MaxInfBR\(k\)NN. We prove that the problem is NP-hard, and develop a non-trivial baseline solution with theoretical guarantees by using state-of-the-art techniques. To support efficient and scalable query processing, we present an efficient batch BR\(k\)NN processing framework, which encompasses effective POI selection policies to offer approximate and heuristic solutions to the problem. Extensive experiments on both real and synthetic datasets demonstrate the effectiveness and efficiency of our proposed methods. In the future, it is of interest to generalize our problem to other popular diffusion models (i.e., the LT model [24], the Time-aware IC model [46]). In addition, considering our problem in the dynamic settings, where the data from the social media evolves over time is also a promising future direction.

Acknowledgements This work was supported in part by the NSFC under Grants No. (62025206, 61972338, and 62102351). Yunjun Gao is the corresponding author of this work.

Author Contribution Pengfei Jin gave the main idea of this paper, programmed the codes and made experiments. The first draft of the manuscript was written by Pengfei Jin and Lu Chen. The material preparation includes figures, data analysis were performed by Pengfei Jin, Xueqin Chang and Zhanyu Liu. Yunjun Gao and Christian S. Jensen gave suggestions and help to improve the manuscript. All the authors read and approved the final manuscript of this paper.

Funding No applicable.

Data availability The codes and corresponding datasets are available at https://github.com/ZJU-DAILY/MaxInfBR\(k\)NN.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for Publication The authors all agree with the content for publication.

Competing interests The authors declare that they have no competing interests.

Human and animal ethics Not applicable.

References

1. Abeywickrama, T., Cheema, M., Khan, A.: K-spin: Efficiently processing spatial keyword queries on road networks. TKDE 32(5), 983–997 (2020)
2. Ahuja, R., Armenatzoglou, N., Papadias, D., Fakas, G.: Geo-social keyword search. In: SSTD, pp. 431–450 (2015)
3. Akiba, T., Iwata, Y., Kawarabayashi, K., Kawata, Y.: Fast shortest-path distance queries on road networks by pruned highway labeling. In: ALENEX, pp. 147–154 (2014)
4. Bian, S., Guo, Q., Wang, S., Yu, J.: Efficient algorithms for budgeted influence maximization on massive social networks. PVLDB 13(9), 1498–1510 (2020)
5. Borgs, C., Brautbar, M., Chayes, J. T., Lucier, B.: Maximizing social influence in nearly optimal time. In: SODA, pp. 946–957 (2014)
6. Cao, X., Cong, G., Guo, T., Jensen, C. S., Ooi, B. C.: Collective spatial keyword querying. In: SIGMOD, p. 373–384 (2011)
7. Chen, L., Cong, G., Jensen, C., Wu, D.: Spatial keyword query processing: an experimental evaluation. PVLDB 6(3), 217–228 (2013)
8. Chen, L., Shang, S., Yang, C., Li, J.: Spatial keyword search: a survey. GeoInformatica, 85–106 (2020)
9. Chen, X., Cao, X., Xu, Z., Zhang, Y., Shang, S., Zhang, W.: Accelerate Maxbrknn Search by Knn Estimation. In: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019, Pp. 1730–1733. IEEE (2019)

10. Chen, X., Deng, L., Zhao, Y., Zhou, X., Zheng, K.: Community-based influence maximization in location-based social network. World Wide WEB 24(6), 1903–1928 (2021)

11. Choudhury, F., Culpepper, J., Bao, Z., Sellis, T.: Finding the optimal location and keywords in obstructed and unobstructed space. VLDBJ 27(4), 445–470 (2018)

12. Choudhury, F., Culpepper, J., Sellis, T., Cao, X.: Maximizing bichromatic reverse spatial and textual k nearest neighbor queries. PVLDB 9(6), 1414–1417 (2016)

13. Domingos, P., Richardson, M.: Mining the network value of customers. In: KDD, pp. 57–66 (2001)

14. Feige, U.: A threshold of $\ln n$ for approximating set cover. J. ACM 20(3), 634–652 (1998)

15. Gkorgkas, O., Vlachou, A., Doulkeridis, C., Norvag, K.: Maximizing influence of spatio-textual objects based on keyword selection. In: SSTD, pp. 413–430 (2015)

16. Guo, Q., Wang, S., Wei, Z., Chen, M.: Influence maximization revisited: efficient reverse reachable set generation with bound tightened. In: SIGMOD, pp. 2167–2181 (2020)

17. Guo, Q., Wang, S., Wei, Z., Chen, M.: Influence Maximization Revisited: Efficient Reverse Reachable Set Generation with Bound Tightened. In: SIGMOD, pp. 62167–2181 (2020)

18. Han, P., Li, Z., Liu, Y., Zhao, P., Li, J., Wang, H., Shang, S.: Contextualized point-of-interest recommendation. In: IJCAI, pp. 2484–2490 (2020)

19. Han, P., Shang, S., Sun, A., Zhao, P., Kai, Z., Kalnis, P.: Auc-Mf: Point of interest recommendation with auc maximization. In: ICDE, pp. 1558–1561 (2019)

20. Huang, K., Tang, J., Han, K., Xiao, X., Chen, W., Sun, A., Tang, X., Lim, A.: Efficient approximation algorithms for adaptive influence maximization. VLDBJ 29(6), 1385–1406 (2020)

21. Hung, H., Yang, D., Lee, W.: Social influence-aware reverse nearest neighbor search. ACM Trans. Spatial Alg. Syst 2(3), 1–35 (2016)

22. Jin, P., Chen, L., Gao, Y., Chang, X., Liu, Z., Jensen, C.S.: Maximizing the influence of bichromatic reverse k nearest neighbors in geo-social networks. [Online] (2022). https://arxiv.org/abs/2204.10203. Accessed 2022-09-18

23. Jin, P., Gao, Y., Lu, C., Zhao, J.: Efficient group processing for multiple reverse top-k geo-social keyword queries. In: DASFAA, pp. 279–287 (2020)

24. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)

25. Kolahdouzan, M., Shahabi, C.: Voronoi-based K nearest neighbor search for spatial network databases. In: VLDB, p. 840–851 (2004)

26. Korn, F., Muthukrishnan, S.: Influence sets based on reverse nearest neighbor queries. In: SIGMOD, pp. 201–212 (2000)

27. Li, G., Chen, S., Feng, J., Tan, K., Li, W.: Efficient location-aware influence maximization. In: SIGMOD, pp. 87–98 (2014)

28. Lu, J., Lu, Y., Cong, G.: Reverse spatial and textual K nearest neighbor search. In: SIGMOD, pp. 349–360 (2011)

29. Lu, Y., Lu, J., Cong, G., Wu, W., Shahabi, C.: Efficient algorithms and cost models for reverse spatial-keyword k-nearest neighbor search. ACM Trans. Database Syst. 39(2), 3:113:46 (2014)

30. Luo, H., Choudhury, F., Culpepper, J., Bao, Z., Zhang, B.: Maxbrknn queries for streaming geo-data. In: DASFAA, pp. 647–664 (2018)

31. Nguyen, H., Dinh, T., Thai, M.: Cost-aware targeted viral marketing in billion-scale. In: INFOCOM, pp. 337–346 (2017)

32. Ohsaka, N., Sonobe, T., Fujita, S., Kawarabayashi, K.: Coarsening massive influence networks for scalable diffusion analysis. In: SIGMOD, pp. 635–650 (2017)

33. Plantadosi, S.: Zipf’s word frequency law in natural language: a critical review and future directions. Psychon. Bull. Rev. 21(5), 1112–1130 (2014)

34. Rao, X., Chen, L., Liu, Y., Shang, S., Yao, B., Han, P.: Graph-flashback network for next location recommendation. In: KDD (2022)

35. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: KDD, pp. 61–70 (2002)

36. Rocha-Junior, J., Norvag, K.: Top-K Spatial Keyword Queries on Road Networks. In: EDBT, pp. 168–179 (2012)

37. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. 24(5), 513–523 (1988)

38. Shang, S., Deng, K., Xie, K.: Best point detour query in road networks. In: GIS, pp. 71–80 (2010)
39. Shang, S., Guo, D., Liu, J., Zheng, K., Wen, J.: Finding regions of interest using location based social media. Neurocomputing 173, 118–123 (2016)
40. Shang, S., Yuan, B., Deng, K., Xie, K., Zhou, X.: Finding the most accessible locations: reverse path nearest neighbor query in road networks. In: GIS, pp. 181–190 (2011)
41. Shang, S., Zheng, K., Jensen, C., Yang, B., Kalnis, P., Li, G., Wen, J.: Discovery of path nearby clusters in spatial networks. TKDE 27(6), 1505–1518 (2015)
42. Shi, J., Mamoulis, N., Wu, D., Cheung, D.: Density-Based Place Clustering in Geo-Social Networks. In: SIGMOD, pp. 99–110 (2014)
43. Tang, J., Tang, X., Xiao, X., Yuan, J.: Online processing algorithms for influence maximization. In: SIGMOD, pp. 991–1005 (2018)
44. Tang, J., Tang, X., Yuan, J.: Influence maximization meets efficiency and effectiveness: a hop-based approach. In: ASNON, pp. 64–71 (2017)
45. Tang, J., Tang, X., Yuan, J.: An efficient and effective hop-based approach for influence maximization in social networks. Soc. Netw. Anal. Min. 8(1), 10–560 (2018)
46. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: a martingale approach. In: SIGMOD, pp. 1539–1554 (2015)
47. Wang, S., Xiao, X., Yang, Y., Lin, W.: Effective indexing for approximate constrained shortest path queries on large road networks. PVLDB 10(2), 61–72 (2016)
48. Wang, X., Zhang, Y., Zhang, W., Lin, X.: Efficient distance-aware influence maximization in geosocial networks, vol. 29 (2017)
49. Wang, Y., Li, J., Zhong, Y., Zhu, S., Guo, D., Shang, S.: Discovery of accessible locations using region-based geo-social data. World Wide WEB 22(3), 929–944 (2019)
50. Wong, R., Ozsu, M., Yu, P., Fu, A., Liu, L.: Efficient method for maximizing bichromatic reverse nearest neighbor. PVLDB 2(1), 1126–1137 (2009)
51. Wu, D., Cong, G., Jensen, C.: A framework for efficient spatial web object retrieval. VLDBJ 21(6), 797–822 (2012)
52. Yao, B., Chen, Z., Xiaofeng, G., Shang, S., Ma, S., Minyi, G.: Flexible aggregate nearest neighbor queries in road networks. In: ICDE, pp. 761–772 (2018)
53. Yu, M.L., Papadias, D., Mamoulis, N., Tao, Y.: Reverse nearest neighbors in large graphs. TKDE 18(4), 540–553 (2006)
54. Zhao, J., Gao, Y., Chen, G., Chen, R.: Why-not questions on top-K geo-social keyword queries in road networks. In: ICDE, pp. 965–976 (2018)
55. Zhao, J., Gao, Y., Chen, G., Jensen, C. S., Chen, R., Cai, D.: Reverse top-K geo-social keyword queries in road networks. In: ICDE, pp. 387–398 (2017)
56. Zhong, R., Li, G., Tan, K.L., Zhou, L., Gong, Z.: G-tree: An efficient and scalable index for spatial search on road networks. TKDE 27(8), 2175–2189 (2015)
57. Zhongpu, C., Bin, Y., ZhiJie, W., Xiaofeng, G., Shuo, S., Shuai, M., Minyi, G.: Flexible aggregate nearest neighbor queries and its keyword-aware variant on road networks. TKDE 33(12), 3701–3715 (2021)

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
Authors and Affiliations

Pengfei Jin\textsuperscript{1} · Lu Chen\textsuperscript{1} · Yunjun Gao\textsuperscript{1} · Xueqin Chang\textsuperscript{1} · Zhanyu Liu\textsuperscript{1} · Shu Shen\textsuperscript{2} · Christian S. Jensen\textsuperscript{3}

Lu Chen
luchen@zju.edu.cn
Yunjun Gao
gaoyj@zju.edu.cn
Xueqin Chang
changxq@zju.edu.cn
Zhanyu Liu
zhanyuliu@zju.edu.cn
Shu Shen
shenshu@huawei.com
Christian S. Jensen
csj@cs.aau.dk

\textsuperscript{1} College of Computer Science, Zhejiang University, Hangzhou 310027, China
\textsuperscript{2} Department of Kunpeng Computing Product, Huawei Technologies CO., LTD, Shenzhen 518129, China
\textsuperscript{3} Department of Computer Science, Aalborg University, Aalborg 410000, Denmark