Efficient Distance-Aware Influence Maximization in Geo-social Networks

Xiaoyang Wang, Ying Zhang, Wenjie Zhang, Xuemin Lin, *Fellow, IEEE*

Abstract—Given a social network $\mathcal{G}$ and a positive integer k, the influence maximization problem aims to identify a set of k nodes in $\mathcal{G}$ that can maximize the influence spread under a certain propagation model. As the proliferation of geo-social networks, location-aware promotion is becoming more necessary in real applications. In this paper, we study the distance-aware influence maximization (DAIM) problem, which advocates the importance of the distance between users and the promoted location. Unlike the traditional influence maximization problem, DAIM treats users differently based on their distances from the promoted location. In this situation, the k nodes selected are different when the promoted location varies. In order to handle the large number of queries and meet the online requirement, we develop two novel index-based approaches, MIA-DA and RIS-DA, by utilizing the information over some pre-sampled query locations. MIA-DA is a heuristic method which adopts the maximum influence arborescence (MIA) model to approximate the influence calculation. In addition, different pruning strategies as well as a priority-based algorithm are proposed to significantly reduce the searching space. To improve the effectiveness, in RIS-DA, we extend the reverse influence sampling (RIS) model and come up with an unbiased estimator for the DAIM problem. Through carefully analyzing the sample size needed for indexing, RIS-DA is able to return a $1 - 1/e - \epsilon$ approximate solution with at least $1 - \delta$ probability for any given query. Finally, we demonstrate the efficiency and effectiveness of proposed methods over real geo-social networks.

Index Terms—Influence maximization, distance-aware, maximum influence arborescence, reverse influence sampling

1 Introduction

With the advance of Web 2.0 techniques and social media platforms, more and more companies are utilizing social networks to promote their products. Influence maximization, which leverages the benefit of word-of-mouth effect in social networks, is a key problem in viral marketing and has been widely studied in the literature [1], [2], [3], [4], [5]. Given a social network $\mathcal{G}$ and a positive integer k, the influence maximization problem aims to identify a set of k nodes, called a seed set, which can maximize the expected number of nodes influenced under a certain propagation model. Since the propagation of influence is based on the trust between families, close friends, etc, this marketing strategy is shown to be more effective than the traditional advertising channels, such as TV and newspapers [6], [7].

The influence maximization problem is formally defined by Kemple et al [4]. In the seminal paper, two models, the independent cascade (IC) model and the linear threshold (LT) model, have been used to describe the propagation of influence over the social network. The authors have proved that the problem is NP-hard under both IC and LT models, and proposed a greedy algorithm with a $1 - 1/e - \epsilon$ approximation ratio based on the monotonic and submodular property of the influence spread functions. The ϵ error is involved due to using the Monte Carlo simulation to calculate the influence spread, which is proved to be #P-Hard.

Motivation. In most existing works concerning the influence maximization problem [1], [2], users in a social network are equally treated, *i.e.,* each user has the same weight. The advance of geo-position enabled devices and services makes it possible to add a spatial dimension to the traditional social networks, *i.e.,* geo-social networks. When conducting a location-aware promotion (*e.g.,* promoting a newly opened restaurant in downtown), existing influence maximization model may not fulfill the requirement because of the ignorance of users' spatial information. Intuitively, users who are close to a promoted location are more likely to attend it. Hence, it is natural to consider that users should be weighted differently with respect to the promoted location. In papers [8], [9], the authors have taken the location information into consideration when selecting the influential seed set. In [9], each user is associated with multiple check-ins, *i.e.,* locations. Based on users' visiting history data, it aims to infer the propagation probability among users given a promoted location, which problem is orthogonal to the study focus in this paper. In [8], each user has a location in 2-dimensional space, and given a query region R, the authors try to select a set of users that will maximize the influence only to the users in R. However, this model has two shortcomings: *1)* Given a promoted location, it is not easy to select a proper target region. *2)* It neglects the importance of distance between users and the promoted location. Following is a motivation example.

Example 1. *As shown in Figure 1, there is a newly opened restaurant "Sokyo" at location q in Sydney. In the figure, the social connections between users are eliminated for the simplicity of explanation. The owner wants to promote his restaurant by using the social network platform, e.g., Facebook. He plans to offer some free meal coupons and*

- *Xiaoyang Wang, Ying Zhang are with University of Technology Sydney, NSW 2007, Australia. Email:{Xiaoyang.Wang, Ying.Zhang}@uts.edu.au*
- *Wenjie Zhang and Xuemin Lin are with School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia. E-mail: {zhangw, lxue}@cse.unsw.edu.au*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2633472, IEEE Transactions on Knowledge and Data Engineering

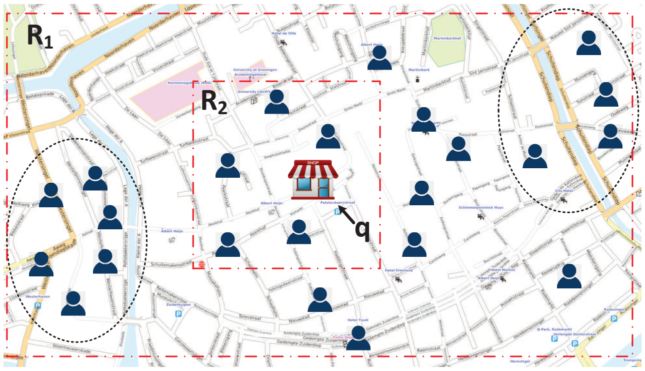


Fig. 1. Motivation Example

VIP cards to a set of influential users, allowing them to propagate the news about the restaurant through the social network. Intuitively, users near the restaurant will become more potential customers, if no other information about users is provided. To select the seed set, i.e., influential users, it will be not easy to employ the location-aware influence maximization model proposed in [8]. This is because, if the query region is too large, e.g., the whole space R_1, it may return a seed set that influences a large number of people near the boundary of R_1, like the users in the two dotted ovals. Even though influenced, these users may not come to the restaurant due to the distance reason. If the query region is too small, e.g., the dotted rectangle R_2, it is possible that only limited users will be influenced and miss a lot of potential users who are not included in the query region. Thus, neither of the cases will be a successful promotion.

As shown in Example 1, it is crucial to consider a user's distance from the promoted location when demonstrating a location-aware promotion through the social network. In this paper, we study the *Distance-Aware Influence Maximization* (DAIM) problem [10] which seamlessly combines the two factors: influence spread and users' distance from the query location. Given a geo-social network $\mathcal{G}$ and a query location q in 2-dimensional space, *i.e.,* the promoted location, each user u in the social network is assigned a weight which is decided by the distance between u and q. The smaller the distance is, the larger the weight of the user will have. In the DAIM problem, the gain of influencing a user is defined as the activated probability multiplied by the user's weight, and the expected influence spread of a seed set is calculated as the weighted influence spread. Under this target function, potential users, that is those close to the promoted location, will be more likely to be influenced. The distance-aware influence maximization problem aims to find a set of k nodes that will maximize the weighted influence.

Challenges. The main challenges of this problem lie in the following two aspects. *Firstly*, the number of nodes to be evaluated is large. Unlike the problem in [8], in the DAIM model, all the users in the social network are considered to be the candidate seeds, the number of which can be very large in real applications. In addition, the calculation of influence spread is time consuming [3], so it is critical to reduce the number of users to be evaluated. *Secondly*, it is not an easy task to meet the online requirement. It is important for the system to support the online requirement, since there may be plenty of DAIM queries issued. Although many works have investigated the influence maximization problem, only few of them can meet the online requirement. In some recent works [8], [11], [12], [13], [14], the authors have attempted to provide index-based algorithms for online service, but their problem settings are different from ours, and it is nontrivial to extend these algorithms to solve the problem proposed in this paper.

To address the above challenges, in this paper, we present two index-based approaches, MIA-DA and RIS-DA. This paper is a journal extension of our previous conference paper [10]. MIA-DA is the method developed in the conference paper, and RIS-DA is the approach developed in the journal extension. MIA-DA extends the maximum influence arborescence (MIA) model [3] to approximate the influence calculation, due to the #P-Hard to compute the influence spread for a set of nodes. Considering that most of nodes in the social network are insignificant (*i.e.,* with small influence), we further develop two pruning rules to avoid computing their influence spread and reduce the searching space. Novel index structures are built to efficiently compute the information needed in the pruning rules. Furthermore, we develop a priority-based algorithm which combines the pruning strategies to find the seed set efficiently. MIA-DA is able to return a result with $1-1/e$ approximation ratio under the MIA model.

However, due to the approximation of the MIA model in MIA-DA approach, the influence of returned seed set may not be satisfied. To improve the effectiveness, in RIS-DA, we utilize the popular reverse influence sampling (RIS) model [15], and come up with an unbiased estimator of influence spread in the DAIM problem. For a given DAIM query, we first analyze the sample size needed to return a $1-1/e-\epsilon$ approximate result with high probability. Then we derive the sample size required for any given query in space based on some pre-sampled queries. Efficient algorithms are provided to index sufficient samples in the offline phase. In the online query processing, we on the fly compute a lower bound for the sample size needed and use the indexed samples to answer the query. For a given query, RIS-DA can return a $1-1/e-\epsilon$ approximate result with at least $1-\delta$ probability.

Contributions. This paper is an journal extension of our previous conference paper [10]. Our principle contributions in the conference paper and journal extension are summarized as follows.

- We formally introduce the distance-aware influence maximization problem over geo-social networks. (In conference paper)
- We propose the MIA-DA approach, which extends the MIA model to support the new problem heuristically. To prune insignificant nodes from the ex-

act calculation, different pruning strategies and a priority-based search algorithm are proposed. (In conference paper)

- We propose the RIS-DA approach, which extends the RIS model and comes up with an unbiased estimator for the DAIM problem. Through analyzing the sample size needed for any potential query, RIS-DA can return a $1-1/e-\epsilon$ approximate result with at least $1-\delta$ probability by using the index. (In journal extension)
- We evaluate the performance of the proposed approaches on real geo-social networks. Our comprehensive experiments confirm the efficiency and effectiveness of the proposed techniques. (In conference paper and journal extension)

Roadmap. The rest of the paper is organized as follows. Section 2 formally introduces the problem and the related techniques used in the paper. Section 3 introduces the details of the MIA-DA approach. In Section 4, we introduce the RIS-DA approach which can return a result with theoretical guarantee. We demonstrate the efficiency and effectiveness of the proposed techniques on real datasets in Section 5. Lastly, we introduce the related works in Section 6 and conclude the paper in Section 7.

2 Background

We first formally introduce the problem of distance-aware influence maximization in Section 2.1. Then we introduce the key techniques utilized in this paper in Section 2.2. Table 1 summarizes the notations frequently used throughout the paper.

Notation	Meaning
$\mathcal{G}=(V,E)$	social/geo-social network
u, v	a node or user in V
$\langle u,v\rangle$	a directed edge from u to v
S	a selected seed set $S \subseteq V$
q	query point in 2-dimensional space
$\mathcal{R}$	a set of RIS samples
l	RIS sample size
θ	threshold for pruning insignificant path
$MIP(u,v)$	maximal influence path between u and v
$d(v,q)$	Euclidean distance between v and q
$w(v,q)$	the weight of node v according to q
$I(S)$ $(I^m(S))$	influence of set S (in MIA)
$I_q(S)$ $(I_q^m(S))$	distance-aware influence spread of S (in MIA)
$I(S,v)$ $(I^m(S,v))$	probability of set S influence v (in MIA)
$I(u\vert S)$ $(I^m(u\vert S))$	marginal influence of u (in MIA)
$I_q(u\vert S)$ $(I_q^m(u\vert S))$	marginal influence of u in DAIM (in MIA)
$I_q^L(\{u\})$, $I_q^U(\{u\})$	lower and upper bound of $I_q^m(\{u\})$
$I_q^U(u\vert S)$	upper bound of $I_q^m(u\vert S)$
OPT_q^k	optimal influence with k nodes for query q
L_q^k	lower bound of OPT_q^k

TABLE 1
The Summary of Notations

2.1 Problem Definition

We consider a geo-social network as a directed graph $\mathcal{G}=(V,E)$, where V represents the set of nodes (users) and E represents the set of edges (relationships between users) in $\mathcal{G}$. Each node $v \in V$ has a geographical location (x,y), where x and y denote the longitude and latitude of v respectively. A function $w: V \times q \rightarrow \mathbb{R}^+$ assigns each node a weight corresponding to a given location q in 2-dimensional space. Given an edge $\langle u,v\rangle \in E$, we say v is an outgoing neighbour of u and u is an incoming neighbour of v.

Diffusion Model. There are many methods to simulate the influence propagation in a social network. In this paper, we focus on the *independent cascade* (IC) model, which is most widely adopted by the existing researches [5], [11], [12], [15]. Under the IC model, each edge $\langle u,v\rangle \in E$ has an independent probability $\Pr(u,v) \in [0,1]$, indicating the probability that u influences v. The influence diffusion of a set $S \subseteq V$ of selected nodes works as follows:

- At timestamp 0, only the nodes in $S_0 = S$ are $active$, while all the other nodes are $inactive$.
- Let S_i denote the set of nodes that are activated at timestamp i. At timestamp $i+1$, each node $u \in S_i$ attempts to activate its each inactive outgoing neighbour v with probability $\Pr(u,v)$.
- Once a node becomes active, it remains activate for the subsequent iterations. The procedure terminates when no more nodes can be activated, *i.e.*, $S_t = \emptyset$, where $t = 0,1,2,\ldots$

The *influence spread* $I(S)$ of S is defined as the expected number of nodes activated by the above procedure, *i.e.*, $I(S) = \mathbb{E}[\sum_{i=0}^{t} S_i]$. The influence spread of S can also be calculated as $\sum_{v\in V} I(S,v)$, where $I(S,v)$ is the probability that S activates v under the IC model.

Definition 1 (Distance-Aware Influence Spread)**.** *Given a geo-social network $\mathcal{G}=(V,E)$ and a promoted location q in 2-dimensional space, the distance aware influence spread of a set of nodes S, denoted as $I_q(S)$, is calculated as $\sum_{v\in V} I(S,v)w(v,q)$, where $w(v,q)$ is the weight of v with respect to q.*

We refer to distance-aware influence spread as influence spread when there is no ambiguity. As discussed in Section 1, users who are close to promoted location should have higher priority to be influenced. Thus $w(v,q)$ should be inversely proportional to the distance between v and q. For ease of exposition, in this paper, we only investigate the case where $w(v,q) = ce^{-\alpha d(v,q)}$, which is a widely used decay function [16]. In the function, $c > 0$ is the maximum weight that a node can achieve, $\alpha > 0$ denotes the decay speed of the node weight, and $d(v,q)$ is the Euclidean distance between v and q. However, the techniques developed in this paper can also be extended to support other distance metrics such as Manhattan distance.

Problem Statement. Given a geo-social network $\mathcal{G}$, a query location q and a positive integer k, the problem of *distance-aware influence maximization* (DAIM) is to find a set S^* of k nodes in $\mathcal{G}$ which has the largest distance-aware influence spread, *i.e.*,

$$S^* = \underset{S \subseteq V}{\arg\max}\{I_q(S)||S|=k\}$$

where S^* is called a seed set and each node $u \in S^*$ is called a seed.

Problem Hardness and Properties. We show the hardness of the problem by considering a simple case when $c = 1$ and $\alpha = 0$ in the function w. In this case, each node's weight equals 1, thus the DAIM problem becomes a traditional influence maximization problem which is NP-hard [4]. Therefore, the DAIM problem is an NP-hard problem. Following a similar routine, we can prove that the computation of influence spread $I_q(S)$ is also a #P-hard problem. To estimation influence spread of S, we can run the Monte Carlo simulations starting from S. As stated in Lemma 1, $I_q(S)$ has the following two properties:

- Monotonic. Given two sets $S,\ T \subseteq V$ and $S \subseteq T$, a function $f(x)$ is monotonic, if $f(S) \leq f(T)$.
- Submodular. Given two sets $S,\ T \subseteq V$, $S \subseteq T$ and $v \in V \setminus T$, a function $f(x)$ is submodular, if $f(T \cup \{v\}) - f(T) \leq f(S \cup \{v\}) - f(S)$.

Lemma 1. *Under the distance-aware influence maximization model, the influence spread function $I_q(S)$ is monotonic and submodular.*

Proof: Let $S,\ T \subset V$, $S \subset T$, and $v \in V \setminus T$. Under the IC model, the influence spread can be computed using the possible world semantics [4]. Specially, $\mathcal{G}$ corresponds to a set of graph instances, *i.e.*, $\{g_i\}$ and $\Pr(g_i)$ denotes the probability of generating g_i. $R_{g_i}(S, v)$ is indicator, where $R_{g_i}(S, v) = 1$ if S can reach v in g_i, *i.e.*, there is a path from a node in S to v in g_i; otherwise $R_{g_i}(S, v) = 0$. Then $I_q(S)$ can be calculated as follows.

$$I_q(S) = \sum_{g_i \in \mathcal{G}} \sum_{u \in V} R_{g_i}(S, u) w(u, q) \Pr(g_i) \tag{1}$$

Monotonic. Since $S \subset T$, for a certain instance g_i, if S can reach u, T must reach u, *i.e.*, $R_{g_i}(T, u) \geq R_{g_i}(S, u)$. Thus, the following equation holds.

$$\begin{aligned} &I_q(T) - I_q(S) \\ =&\sum_{g_i \in \mathcal{G}} \sum_{u \in V} (R_{g_i}(T, u) - R_{g_i}(S, u)) w(u, q) \Pr(g_i) \geq 0 \end{aligned}$$

Therefore, the monotonic property is proved.

Submodular. Given a node u and an instance g_i, if v can reach u in g_i, $R_{g_i}(S \cup \{v\}, u) = R_{g_i}(T \cup \{v\}, u) = 1$. Then we have

$$\begin{aligned} &R_{g_i}(T \cup \{v\}, u) - R_{g_i}(T, u) - (R_{g_i}(S \cup \{v\}, u) \\ &- R_{g_i}(S, u)) \\ =&R_{g_i}(S, u) - R_{g_i}(T, u) \leq 0 \end{aligned} \tag{2}$$

Otherwise, *i.e.*, v cannot reach u, we have

$$\begin{aligned} &R_{g_i}(T \cup \{v\}, u) - R_{g_i}(T, u) - (R_{g_i}(S \cup \{v\}, u) \\ &- R_{g_i}(S, u)) \\ =&R_{g_i}(T, u) - R_{g_i}(T, u) - (R_{g_i}(S, u) - R_{g_i}(S, u)) = 0 \end{aligned} \tag{3}$$

By combining Equation (1), (2) and (3), we have

$$\begin{aligned} &I_q(T \cup \{v\}) - I_q(T) - (I_q(S \cup \{v\}) - I_q(S)) \\ =&\sum_{g_i \in \mathcal{G}} \sum_{u \in V} (R_{g_i}(T \cup \{v\}, u) - R_{g_i}(T, u) - (R_{g_i}(S \cup \{v\}, u) \\ &- R_{g_i}(S, u))) w(u, q) \Pr(g_i) \leq 0 \end{aligned}$$

Therefore, the submodular property is proved. □

Therefore, we can use the greedy algorithm to incrementally select the node with the largest marginal gain. Algorithm 1 shows the details of the method. The greedy algorithm can return a result with $1 - 1/e - \epsilon$ approximation ratio, where ϵ is the error generated by using the Monte Carlo simulation to estimate the influence spread.

Algorithm 1: Naive Greedy Algorithm

```
Input  : $\mathcal{G}$ : a geo-social network, $k$ : seed set size,
         $q$ : query location.
Output : $S$ : a set of $k$ nodes
$S \leftarrow \emptyset$ ;
while $|S| < k$ do
    $u \leftarrow \arg\max_{w \in V \setminus S} (I_q(S \cup \{w\}) - I_q(S))$ ;
    $S \leftarrow S \cup \{u\}$ ;
return $S$
```

2.2 Preliminary

The MIA model and the RIS model are designed for solving the traditional influence maximization problem. In this paper, we develop two approaches by extending these two models respectively. In this section, we introduce the details of the two models.

2.2.1 MIA Model

To solve the influence maximization problem, a fundamental step is to calculate the influence spread for a set of nodes. Due to the hardness of calculating influence spread, the MIA model [3] utilizes a tree-based heuristic to approximate it. The idea of the MIA model is described as follows.

Given a social network $\mathcal{G}$ and two nodes $u, v \in V$, u can activate v if there is a path between the two nodes. A path between u and v is denoted as $p(u, v) = \langle u = w_1, w_2, ..., w_m = v \rangle$ where $\langle w_i, w_{i+1} \rangle \in E$ and $i = 1, 2, \ldots, m - 1$. The probability that u will activate v through $p(u, v)$ is calculated as $\Pr(p(u, v)) = \prod_{i=1}^{m-1} \Pr(w_i, w_{i+1})$. Under the MIA model, when there are multiple paths between u and v, the path with the largest probability is selected, because this path presents the greatest opportunity for u to influence v. The path with the largest probability between u and v is called the maximal influence path, denoted as $MIP(u, v)$, *i.e.*,

$$MIP(u, v) = \underset{p \in p(u, v, \mathcal{G})}{\arg\max} \{\Pr(p)\} \tag{4}$$

where $p(u, v, \mathcal{G})$ denotes all the paths between u and v in $\mathcal{G}$. Under the MIA model, u can influence v only through $MIP(u, v)$. However, the probability of many maximal influence paths is quite small, thus it is insignificant to the contribution of measuring the influence spread. We use a threshold θ to prune all the insignificant paths, that is if $\Pr(MIP(u, v)) < \theta$, u is not able to activate v. Hereafter in this paper, we denote $MIP(u, v)$ as the maximal influence path between u and v with probability not smaller than θ, otherwise there is no maximal influence path between u and v. To calculate the influence from S to v, we utilize the *Maximum Influence In(Out) Arborescence* structure [3], which assembles the maximal influence paths to(from) a node v.

Definition 2 (Maximum Influence In(Out) Arborescence)**.** *Given a social network G, the Maximum Influence In Arborescence of a node v, denoted as $MIIA(v)$, is:*

$$MIIA(v) = \cup_{u \in V} MIP(u, v)$$

The Maximum Influence Out Arborescence of v, denoted as $MIOA(v)$, is:

$$MIOA(v) = \cup_{u \in V} MIP(v, u)$$

$MIIA(v)$ and $MIOA(v)$ are trees rooted at v, including all the nodes that can influence and be influenced by v in $\mathcal{G}$. To influence v, S must firstly influence the neighbors $N^m(v)$ of v in $MIIA(v)$. Note that $N^m(v)$ does not equal the incoming neighbors of v in $\mathcal{G}$. If $v \in S$, then $I^m(S, v) = 1$. By traversing $MIIA(v)$ from v to the leaf nodes, we can calculate $I_q(S, v)$ as shown in Equation (5).

$$I^m(S, v) = 1 - \prod_{w \in N^m(v)} (1 - \Pr(S, w, v)\Pr(w, v)) \quad (5)$$

where $\Pr(S, w, v)$ is the probability that S will influence w through $MIIA(v)$ and $\Pr(S, w, v) = 1$ when $w \in S$. Through MIA approximation, we calculate the influence of a set of nodes in polynomial time.

With MIA approximate, the influence maximization problem is still NP-Hard, but the objective function $I^m(S)$ is submodular and monotonic, thus a greedy algorithm can return a result with $1 - 1/e$ approximation ratio under the MIA model. In this paper, we extend the MIA model to support DAIM problem, and use different pruning strategies to speedup the search.

2.2.2 RIS Model

The reverse influence sampling (RIS) model is first proposed in [15] for the traditional influence maximization problem. Before introducing the RIS model, we fist clarify some concepts for the ease of explanation.

Graph Distribution. The social network $\mathcal{G}$ under a probabilistic diffusion model $\mathcal{M}$ can be treated as a graph distribution $\mathcal{G} = \{g_i\}$. Each instance g_i in the distribution corresponds to a deterministic graph generated by following the diffusion model $\mathcal{M}$. In the IC model, g_i is generated by flipping a coin on each edge $\langle u, v\rangle$, which is survived with probability $\Pr(u, v)$.

Definition 3 (Reverse Reachable Set)**.** *Given an instance g_i from $\mathcal{G}$ and a node v, the reverse reachable (RR) set is the set of nodes that can reach v in g_i.*

Definition 4 (Random Reverse Reachable Set)**.** *An RR set is a random reverse reachable set if g_i is randomly sampled from $\mathcal{G}$ and v is randomly selected from all the nodes V.*

A random reverse reachable set is denoted as R_i. In the paper hereafter, we use RR set to denote random RR set for short if there is no ambiguity. We also call a RR set R_i as a RIS sample, and the random selected node v as the sampled node. The intuition of the reverse influence sampling model is that given a set $\mathcal{R}$ of RR sets, if S appears in $\mathcal{R}$ frequently, then it is more likely to have a larger influence spread. Equation 6 is an unbiased estimation of S's influence spread, where $S \cap \mathcal{R}$ is the set of samples that are covered by S, *i.e.*, S can reach the corresponding sampled nodes.

$$\hat{I}(S) = n \times \frac{|S \cap \mathcal{R}|}{|\mathcal{R}|} \quad (6)$$

In consequence, we can sample a set of sufficient samples, and greedily select k nodes with the largest marginal coverage over the samples. Since the selection cost is linear to the number of samples, the major problem in the RIS model is to decide a proper sample size. In [15], the authors mainly focus on the theoretical analysis and renders a large constant factor in the equation of sample size. In [1], [2], Tang et al. significantly reduce the sample complexity and make the model working in practice. As shown in [1], if the sample size l fulfills Equation (7), the greedy algorithm on the RIS model can return a result with $1 - 1/e - \epsilon$ approximation ratio of high probability, where α, β are values decide by the input graph and error parameters, and OPT is the influence spread of the optimal seed set.

$$l \geq \frac{2n \cdot ((1 - 1/e) \cdot \alpha + \beta)^2}{OPT \cdot \epsilon^2} \quad (7)$$

In this paper, we extend the RIS model to support DAIM and return a result with theoretical guarantee.

3 MIA-DA Approach

In this section, we present the MIA-based method, MIA-DA, which speedup the DAIM problem processing by using the MIA heuristic. In Section 3.1, we show how to use MIA model to approximate the computation. Section 3.2 demonstrates how to derive the information for the pruning rules and the index structures. Finally, we present the search algorithm in Section 3.3.

3.1 MIA-based Approximation For DAIM

Based on the idea of the MIA model, we can approximate the influence calculation by multiplying the influence calculated with the weight of node, *i.e.*, $I^m_q(S, v) = I^m(S, v)w(v)$. The influence spread of S can be calculated as follows.

$$I^m_q(S) = \sum_{v \in V} I^m(S, v)w(v) \quad (8)$$

After relaxing the influence calculation with the MIA model, we can compute $I^m_q(S)$ efficiently. However, the problem of finding a set of k nodes that will maximize the influence spread is still NP-hard as stated in Lemma 2 with proof in Appendix A.

Lemma 2. *Under the MIA model, the problem of solving the DAIM problem is NP-hard.*

In addition, the monotonic property and submodular property of the object function still hold. In hence, we can modify the naive greedy algorithm in Algorithm 1 by replacing influence calculation by using the MIA model. Then the algorithm can return a result with approximation ratio $1 - 1/e$ under the MIA model. In greedy algorithm, we pre-compute the $MIIA(v)$ and $MIOA(v)$ offline for each node $v \in V$, because there may be many queries raised and the structures are repeatedly used. For each iteration, we select the node u with maximal marginal influence $I^m_q(u|S)$ under the MIA model. Chen et al. [3] have provided an efficient approach for calculating the marginal influence for all nodes.

3.2 Pruning Rules and Index Structures

The main limitation of the greedy algorithm is that it is required to compute the influence or marginal influence for many nodes. Usually, most of these nodes are insignificant, *i.e.,* they have small influence. Thus we should avoid conducting the exact evaluation for these nodes. Suppose we have an oracle which allows us to achieve the upper bound $I^U_q(\{v\})$ and lower bound $I^L_q(\{v\})$ of each node's influence, and the upper bound $I^U_q(v|S)$ of each node's marginal influence with little cost, then we can immediately devise the following two rules to prune the insignificant nodes from further evaluation.

Rule 1. For the first seed selection, if $I^m_q(\{u\}) \geq I^U_q(\{v\})$ or $I^L_q(\{u\}) \geq I^U_q(\{v\})$, we can directly prune v from the first seed selection.

Rule 2. For subsequent seed selection, if $I^m_q(u|S) \geq I^U_q(\{v\})$ or $I^m_q(u|S) \geq I^U_q(v|S)$, then v can be pruned from the current iteration's evaluation.

In this paper, we investigate the techniques for obtaining the bounds used in the above pruning rules. The idea is that we offline sample a set of points and assume them as the query locations. Then we pre-compute the influence of nodes based on these sampled locations and store them in the index. Given a query location, we can use the closest sampled location to efficiently derive the bounds needed for online queries. For nodes with larger influence, we further partition the space for them to derive tighter bounds. The detailed techniques of deriving the bounds can be found in Appendix B and C or in our conference paper [10].

3.3 Search Algorithm

For the search algorithm, it consists two parts, offline index construction and online query processing. In the offline phase, we build the index used by Rule 1 and 2. For online query processing, we use a priority-based method, that is we explore the nodes based on their influence upper bound. We continuously explore the nodes, and prune or verify them based on the two rules. If we cannot prune or verify a node, we then compute their exact marginal influence. A node is selected into the seed in an iteration, if its marginal influence or lower bound of its marginal influence is larger than all the other nodes' marginal influence's upper bound. The algorithm terminates when k nodes are selected. The details of the algorithm can be found in Appendix D or refer to the conference paper [10].

4 RIS-DA Approach

In the MIA-DA approach, we can efficiently identify a set of k nodes by using the pruning rules and the novel search algorithm. In addition, since it builds the index for all the nodes, it is easy to adopt new constraints over the selected nodes, such as the problem in [17]. In [17], each node u in the social network is associated with a set $\mathcal{A}(u)$ of keywords (*e.g.,* abilities, interests). The authors define a *influential cover set* problem. Given a set Q of keywords and an integer k, it aims to find a k-node set S, which covers the keywords Q, *i.e.,* $Q \subseteq \cup_{u \in S}\mathcal{A}(u)$, and the influence of S is maximized. Even though MIA-DA can return a result with $1-1/e$ approximation ratio under the MIA model, the result may not be as good as the seed set returned by the naive greedy method in Algorithm 1. This is because in MIA-DA, the influence of a node is approximated with the MIA model, which may be deviated from the true influence of the node. In Algorithm 1, the influence is computed using the Monte Carlo simulation, which is an unbiased estimation, and it can return a result with theoretical guarantee for any input graphs. This can also be observed from the experiment results compared with the RIS-DA method proposed in this section. Like Algorithm 1, RIS-DA can offer the same theoretical guarantee, and RIS-DA reports higher influence spread than MIA-DA in the experiments.

In this section, we extend the RIS model to support the DAIM problem. In Section 4.1, we come up with an unbiased estimator for influence spread and the algorithm for solving DAIM problem when a set of RIS samples is given. In Section 4.2, we carefully analyze the number of samples needed to solve a certain DAIM query with theoretical guarantee. In Section 4.3, we present the method to obtain the number of samples needed for offline index. Based on the indexed samples, we can answer any potential query with theoretical grantee in the online processing phase.

4.1 RIS-based Estimation for DAIM Problem

As stated in Section 2.2.2, the RIS model is designed for the traditional influence maximization problem, where each node has the same weight. In hence, the RIS model cannot be directly applied for the DAIM problem.

To tackle this problem, we can take the weight of each sampled node (*i.e.,* the source node of the sample) into account. In traditional influence maximization problem, when estimating the influence of a given node set S, the event of S covering a given sample follows the $\{0,1\}$-Bernoulli distribution. Given a set $\mathcal{R}$ of samples, we use random variable

X_i for each sample. With $\frac{I(S)}{n}$ probability $X_i = 1$, *i.e.*, the sample is covered by S, and with $1 - \frac{I(S)}{n}$ probability $X_i = 0$, *i.e.*, S does not cover the sample. Thus $\frac{|S \cap \mathcal{R}|}{|\mathcal{R}|}$ is an unbiased estimation of $\frac{I(S)}{n}$, *i.e.*, $\mathbb{E}[X_i] = \frac{I(S)}{n}$. By extending this idea, given a query location q and a set of samples, we use the random variable Y_i for each sample in the DAIM problem, and v_i is the corresponding sampled node. For a given set S, with $\frac{I(S)}{n}$ probability $Y_i = w(v_i, q)$ and with $1 - \frac{I(S)}{n}$ probability $Y_i = 0$. Therefore, given a set $\mathcal{R}$ of samples, we can use Equation (9) to estimate $I_q(S)$ and according to Lemma 3, it is an unbiased estimator.

$$\hat{I}_q(S) = n \times \frac{\sum_{R_i \in \mathcal{R} \cap S} w(v_i, q)}{l} \tag{9}$$

where $\hat{I}^r_q(S)$ is the estimator for $I_q(S)$, R_i is a sample covered by S and v_i is the corresponding sampled node. $l = |\mathcal{R}|$ is the sample size.

Lemma 3. *Given a set $\mathcal{R}$ of random RR sets, Equation* (9) *is an unbiased estimation of distance-aware influence spread of S.*

Proof: We follow the procedure in [18] to prove the correctness of the lemma. Let g be an instance of $\mathcal{G}$ in the IC model and $R_g(v)$ denotes the set of node that can reach v in instance g. Then we have the following equation based on the definition of distance-aware influence spread.

$$\begin{aligned} I_q(S) &= \sum_{v \in V} I(S, v) \times w(v, q) \\ &= \sum_{v \in V} \Pr_{g \sim \mathcal{G}}[\exists u \in S \text{ } and \text{ } u \in R_g(v)] \times w(v, q) \\ &= n \times \Pr_{v, g \sim \mathcal{G}}[\exists u \in S \text{ } and \text{ } u \in R_g(v)] \times w(v, q) \\ &= n \times \mathbb{E}[Y_i] \end{aligned}$$

In addition, we have:

$$\mathbb{E}(\hat{I}_q(S)) = \mathbb{E}[n \times \frac{\sum_{R_i \in \mathcal{R} \cap S} w(v_i, q)}{l}] = n \times \mathbb{E}[Y_i]$$

Thus, the correctness of the lemma is proved. □

Given the unbiased estimation of influence spread and a set of samples $\mathcal{R}$, we can use the greedy algorithm to find the seed set. The detail of the method is presented in Algorithm 2.

In Algorithm 2, we first build a bipartite over the set of samples in Line 2, which denotes the coverage relationship between the nodes in V and the sampled nodes. For each node v in $V \setminus S$, $Score[v]$ denotes the weight sum of the sampled nodes that are covered by v but not covered by S. Initially, S is empty, so $Score[v]$ equals the weight sum of the sampled nodes covered by v (Line 3). Then we iteratively select the node u from $V \setminus S$ which has the largest marginal gain, *i.e.*, the one with the largest value in $Score$ (Line 5), and add it into S. Next, we update the $Score$ value for the rest nodes from Line 6 to 9, since some samples are covered by the selected node u. $R_i(v_i)$ is the set of nodes that can return the sampled node v_i from R_i. The algorithm terminates, until k nodes have been selected.

Algorithm 2: RIS-based Greedy Algorithm for DAIM

```
Input  : $\mathcal{G}$ : a geo-social network, $k$ : seed set size, $q$ :
         query location, $\mathcal{R}$ : a set of samples.
Output : $S$ : the seed set of size $k$
1  $S, Score \leftarrow \emptyset$ ;
2  Build a bipartite over $\mathcal{R}$ ;
3  $Score[u] \leftarrow \sum_{R_i \in \{u\} \cap \mathcal{R}} w(v_i, q)$ ;
4  while $|S| < k$ do
5      $u \leftarrow \arg\max_{w \in V \setminus S}\{Score[w]\}$ ;
6      for each $R_i \in \{u\} \cap \mathcal{R}$ do
7          if $R_i \in S \cap \mathcal{R}$ then
8              continue ;
9          for each $u' \in R_i(v_i)$ do
10             $Score[u'] = Score[u'] - w(v_i, q)$ ;
11     $S \leftarrow S \cup \{u\}$;
12 return $S$
```

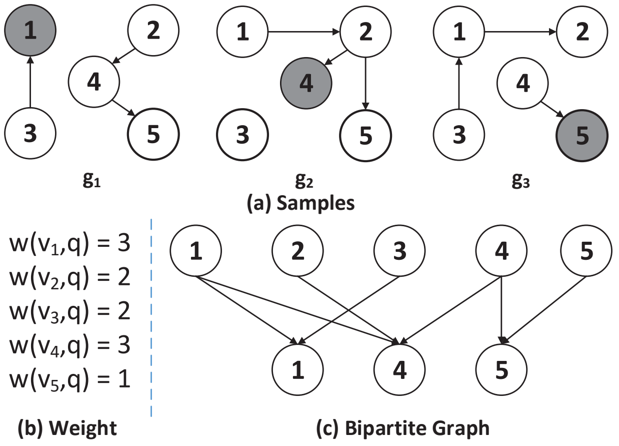


Fig. 2. Example for RIS-DA

Example 2. *Given the query q of Figure 9 in Appendix A, the node weights are shown in Figure 2(b). We randomly sample 3 samples. The sampled instances are shown in Figure 2(a) and the sampled nodes are the dark nodes in the instance. Note that, to obtain a sample, we do not need to materialize the entire instance g_i. We only need to materialize the part that can reach the sampled node. Here, we materialize the entire instances just for the simplicity of understanding.*

Given the three samples, the corresponding bipartite graph is shown in Figure 2(c). The upper layer consists of the nodes in V and the lower layer consists of all the sampled nodes. The edge denotes that the nodes can reach the sampled nodes. For example, nodes v_1 and v_3 can reach the sampled node in g_1, so the sampled node has two incoming edges. Based on the bipartite graph, we can compute the initial $Score$ for each node, which are 6, 3, 3, 4, 1 respectively. Thus node v_1 is the first node selected.

4.2 Sample Size Analysis

In this section, we analyze the sample size required to return a $1 - 1/e - \epsilon_0$ approximate result with at least $1 - \delta_0$ probability by using Algorithm 2. In our analysis, we use the Chernoff bound below:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2633472, IEEE Transactions on Knowledge and Data Engineering

Lemma 4. *Let X be the sum of b i.i.d random variable for samples from [0,1] with sample mean equal to μ. Then for any $\epsilon' > 0$, we have*

$$Pr[X - b\mu \geq \epsilon' \cdot b\mu] \leq \exp\left(-\frac{\epsilon'^2 \cdot b\mu}{2+\epsilon'}\right)$$

$$Pr[X - b\mu \leq -\epsilon' \cdot b\mu] \leq \exp\left(-\frac{\epsilon'^2 \cdot b\mu}{2}\right)$$

To determine the sample size required, we divide the proof procedure into three phases. In Lemma 5, we show that the estimated influence spread of the the selected node set by using Algorithm 2 is close to the optimal influence with bounded approximation ratio of high probability. Then in Lemma 6, we show that when Lemma 6 holds, the returned seed set has $1-1/e-\epsilon_0$ approximation ratio with high probability. By combining the two lemmas, we demonstrate the sample size needed for answering a DAIM query with theoretical guarantee in Lemma 7. Given a DAIM query, *i.e.*, a query location q and a positive integer k, let OPT_q^k be the influence spread of the optimal seed set with k nodes and $w_{\max}$ is the maximum weight among all the nodes. In our problem setting, $w_{\max}$ equals c. Given a DAIM query, let S_q^k denote the seed set returned by Algorithm 2 and S_q^{k*} denote the seed set with optimal influence, *i.e.*, $\mathbb{E}[\hat{I}_q(S_q^{k*})] = OPT_q^k$. Then we have the following lemma hold.

Lemma 5. *Given a DAIM query, $\epsilon_1 > 0$ and $\delta_1 \in (0,1)$, if the sample size l_1 fulfills the following equation:*

$$l_1 = \frac{2nw_{\max} \cdot \log(\delta_1^{-1})}{\epsilon_1^2 OPT_q^k}$$

Then we have the following equation holds with at least $1-\delta_1$ probability.

$$\hat{I}_q(S_q^k) > (1-1/e)(1-\epsilon_1)OPT_q^k \tag{10}$$

Proof: Let random variable $Z_i = \frac{w(v_i,q)}{w_{\max}}$ if S_q^{k*} covers the sample, otherwise $Z_i = 0$. Thus $nw_{\max}\mathbb{E}[Z_i] = OPT_q^k$, and $\hat{I}_q(S_q^{k*}) = n \cdot w_{\max} \cdot \frac{\sum Z_i}{l_1}$. Since each node's weight is within $[0, w_{\max}]$, $Z_i \in [0,1]$. Then we have:

$$\begin{aligned}
&\Pr[\hat{I}_q(S_q^{k*}) \leq (1-\epsilon_1) \cdot OPT_q^k] \\
=&\Pr[\frac{nw_{\max}}{l_1} \cdot \sum Z_i \leq (1-\epsilon_1) \cdot nw_{\max} \cdot \mathbb{E}[Z_i]] \\
=&\Pr[\sum Z_i - l_1\mathbb{E}[Z_i] \leq -\epsilon_1 \cdot l_1\mathbb{E}[Z_i]] \\
\leq&\exp(-\frac{\epsilon_1^2}{2} \cdot l_1\mathbb{E}[Z_i]) \\
=&\exp(-\frac{\epsilon_1^2}{2} \cdot \frac{OPT_q^k l_1}{nw_{\max}}) \\
\leq&\delta_1
\end{aligned}$$

Since S_q^k is the seed set selected by using Algorithm 2, then it should be a $1-1/e$ approximate result over the weighted maximum coverage problem. Then we have the following equation hold with at least $1-\delta_1$ probability.

$$\begin{aligned}
\hat{I}_q(S_q^k) &\geq (1-1/e)\hat{I}_q(S_q^{k*}) \\
&\geq (1-1/e)(1-\epsilon_1)OPT_q^k
\end{aligned}$$

Therefore, the lemma is correct. □

Lemma 6. *Given a DAIM query q, k, let $\epsilon_0, \epsilon_1, \epsilon_2 > 0$, $\epsilon_2 = \epsilon_0 - \epsilon_1(1-1/e)$ and $\delta_2 \in (0,1)$. If Equation (10) holds and the sample size l_2 satisfies the following equation:*

$$l_2 = \frac{2(1-1/e)nw_{\max}\log\binom{n}{k}/\delta_2}{OPT_q^k\epsilon_2^2}$$

Then, the following equation holds with at least $1-\delta_2$ probability:

$$I_q(S_q^k) > (1-1/e-\epsilon_0) \cdot OPT_q^k \tag{11}$$

Proof: For each sample R_i and the corresponding sampled node v_i, we set random variable $Z_i = \frac{w(v_i,q)}{w_{\max}}$, if v_i is covered by a node set S_q^k. Otherwise $Z_i = 0$. Let $\mu = I_q(S_q^k)$ and $a = \frac{\epsilon_2}{\mu}OPT_q^k$. Since Equation (10) holds, we have:

$$\begin{aligned}
\hat{I}_q(S_q^k) &> (1-1/e)(1-\epsilon_1)OPT_q^k \\
&= (1-1/e)OPT_q^k - \epsilon_1(1-1/e)OPT_q^k \\
&= (1-1/e-\epsilon_0)OPT_q^k + \epsilon_2 OPT_q^k
\end{aligned}$$

Based on Chernoff bound, we demonstrate the probability of event $I_q(S_q^k) \leq (1-1/e-\epsilon_0)OPT_q^k$ happens.

$$\begin{aligned}
&\Pr[I_q(S_q^k) \leq (1-1/e-\epsilon_0)OPT_q^k] \\
\leq&\Pr[\hat{I}_q(S) - I_q(S) \geq \epsilon_2 \cdot OPT_q^k] \\
=&\Pr[\sum Z_i - \frac{l_2}{nw_{\max}}\mu \geq \frac{l_2}{nw_{\max}}\mu \cdot \frac{\epsilon_1}{\mu}OPT_q^k] \\
\leq&\exp(-\frac{a^2}{2+a} \cdot \frac{l_2}{nw_{\max}}\mu) \\
\leq&\exp(-\frac{\epsilon_2^2 OPT_q^{k^2}}{2\mu + \epsilon_2 OPT_q^k} \times \frac{l_2}{nw_{\max}}) \\
\leq&\exp(-\frac{\epsilon_2^2 OPT_q^{k^2}}{2(1-1/e-\epsilon_0)OPT_q^k + \epsilon_2 OPT_q^k} \times \frac{l_2}{nw_{\max}}) \\
\leq&\exp(-\frac{\epsilon_2^2 OPT_q^k}{2(1-1/e)} \times \frac{l_2}{nw_{\max}}) \leq \delta_2/\binom{n}{k}
\end{aligned}$$

Since there are at most $\binom{n}{k}$ sets with k nodes, by applying union bound, the lemma is correct. □

Based on Lemma 5 and 6, we can derive the sample size required as shown in Lemma 7.

Lemma 7. *Given a social network $\mathcal{G}$ and a DAIM query, let $\epsilon_0, \epsilon_1, \epsilon_2, \delta_1, \delta_2$ be the parameters defined in Lemma 5 and 6, and $\delta_1 = \delta_2 = \delta_0/2$. If the sample size $l_0 = l_1$ and ϵ_1 fulfills Equation (12), Algorithm 2 can return a $1-1/e-\epsilon_0$ approximate result with at least $1-\delta_0$ probability.*

$$\epsilon_1 = \frac{\epsilon_0\sqrt{\log 2/\delta_0}}{(1-1/e)\sqrt{\log 2/\delta_0} + \sqrt{(1-1/e)\log 2\binom{n}{k}/\delta_0}} \tag{12}$$

Proof: When ϵ_1 fulfills Equation (12), $l_0 = l_1 = l_2$, thus both Lemma 5 and 6 hold. Since the correctness of Lemma 6 is based on Equation (10), $I_q(S_q^k) > (1-1/e-\epsilon_0) \cdot OPT_q^k$ holds with at least $1-\delta_1-\delta_2 = 1-\delta_0$ probability. Thus, the lemma is correct. □

For the simplicity, we use function $l(\epsilon_0, \delta_0, q, k, L_q^k)$ to denote the sample size requirement in Lemma 7, where ϵ_0, δ_0 are the input parameters, q, k corresponds to a DAIM query, and L_q^k is a lower bound of OPT_q^k. When $L_q^k = OPT_q^k$, $l(\epsilon_0, \delta_0, q, k, L_q^k)$ equals l_0 in Lemma 7, otherwise $l(\epsilon_0, \delta_0, q, k, L_q^k) > l_0$.

4.3 Index Construction

In this section, we present the method to index enough samples for online processing. Given Algorithm 2 and the sample size requirement in Lemma 7, we still cannot answer the DAIM problem for online requirement, because of the following two factors:

- To get the sample size required, we need to derive a lower bound of OPT_q^k, which is not a trivial task. Firstly, the DAIM problem is NP-Hard. In addition, we cannot simply use k to sever as a lower bound of OPT_q^k, because OPT_q^k can be smaller than k in extreme case. For example, when all the nodes are far from the query location, the sum of all the nodes' weight can be smaller than k. Most importantly, we hope to derive a tight lower bound efficiently, in order to reduce the computation cost and sample size.
- To get the sufficient number of samples for indexing is not easy. Suppose we can derive a tight lower bound of OPT_q^k efficiently, it is still not satisfied for online requirement. Since when query location varies, OPT_q^k changes correspondingly. In addition, the number of query locations is infinite, so we cannot afford to enumerate all the queries to determine the sample size. Moreover, we do not want to do sampling in the online phase, which may delay the query processing. In hence, it is critical to decide a sample size to index offline, which is able to answer any potential queries with theoretical guarantee.

Considering the above two factors, we can define the the sufficient sample size as follow.

Definition 5 (Sufficient Sample Size for DAIM Query)**.** *Given a geo-social network $\mathcal{G}$ and $\epsilon > 0, \delta \in (0, 1)$, we offline sample a set $\mathcal{R}$ of samples. Given any $k \in [1, k_{\max}]$ and any query location in the space, we say the size of $\mathcal{R}$ is sufficient if we can return a $1 - 1/e - \epsilon$ result with $1 - \delta$ probability by only using $\mathcal{R}$.*

$k_{\max}$ is the maximum value of k allowed for a DAIM query. This setting is quite natural in real applications. For example, the users only allow to select 100 seeds at most.

For any query location q and $k \in [1, k_{\max}]$, suppose we can derive a lower bound L_q^k of OPT_q^k with at least $1 - \delta_0$ probability. If the indexed sample size equals $l(\epsilon, \delta - \delta_0, q, k, L_q^k)$, Algorithm 2 can return a $1 - 1/e - \epsilon$ with $1 - \delta$ probability by applying union bound. In order to index sufficient samples, we need to determine a L_q^k for each query q. The idea is that we first compute the DAIM problem for a set P of sampled query locations, called pivots. Then using the influence computed over these pivots, we can derive the lower bound of OPT_q^k for any query location. In Section 4.3.1, we present an efficient algorithm to compute the DAIM problem for each sampled pivot. In Section 4.3.2, we present the method to derive a lower bound of OPT_q^k for all the queries based on the influence computed on the pivots. Finally, we show the method to index sufficient samples in Section 4.3.3.

4.3.1 Compute Pivot Information

Each pivot $p \in P$ is a location in 2-dimensional space. By consider p as a query location, to return a result with $1 - 1/e - \epsilon_0$ approximation ratio of $1 - \delta_0$ probability, we need to obtain a lower bound of OPT_p^k to compute the sample size required. Naively, we can use the top-k weight nodes and use the sum of their weight as a lower bound. This lower bound may be too passive and results a long offline computation time. In this paper, we use a heuristic method to quickly compute a lower bound for OPT_p^k. The details of the method is presented in Algorithm 3.

This method only considers the two-hop neighbours of S and we only compute the influence of S to these nodes through the pathes with length less than 3. This method can be effective, since the influence decreases quickly with the length of path. Calculating the exact influence spread is #P-Hard, while only considering the two-hop neighbours can be very efficient. Intuitively, a node with higher out degree and larger weight will be more promising to sever as seeds. Thus, we first sort all the nodes based on $w(v, q)|N_{out}(v)|/w_{\max}$. We use the k nodes with largest score as the selected nodes (Line 2). Q is a vector, which stores the two-hop neighbours of S (Line 3). For each node $v \in Q$, we compute the influence from S to v only through pathes with length less than 3. Clearly, the algorithm return a lower bound of OPT_p^k with 100% probability.

Algorithm 3: LowerBoundEstimation

```
  Input  : $\mathcal{G}$ : a social network; $k$ : number of seeds; $p$ :
           a pivot.
  Output : $L_p^k$ : lower bound of $OPT_p^k$
1 Sort nodes based on $w(v, p)|N_{out}(v)|/w_{\max}$ ;
2 $S \leftarrow$ top-$k$ nodes ;
3 $Q \leftarrow S$ two-hop neighbours not include $S$ ;
4 $L_p^k \leftarrow \sum_{v \in S} w(v, p)$ ;
5 for each $v \in Q$ do
6     $L_p^k += $ influence from $S$ to $v$ through pathes with
      length less than 3 ;
7 return $L_p^k$
```

Example 3. *As shown in Figure 9 in Appendix A, suppose $k = 1$. If v_2 is the selected node in Algorithm 3, then v_4 and v_5 are its two-hop neighbours. The influence from S to v_4 is calculated as $0.5 \times w(v_4, p)$ and for v_5, the influence is calculated as $(1 - (1 - 0.5)(1 - 0.5 \times 0.5)) \times w(v_5, p) = 0.625 \times w(v_5, p)$. If v_3 is the selected node, the two-hop neighbours include all the other nodes. When computing the influence from v_3 to v_5, we only consider the path $\langle v_3, v_4, v_5 \rangle$, since both $\langle v_3, v_1, v_2, v_5 \rangle$ and $\langle v_3, v_1, v_2, v_4, v_5 \rangle$ are pathes with length larger than 2.*

Given Algorithm 3, we only need to sample enough samples for each pivot and compute the influence.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2633472, IEEE Transactions on Knowledge and Data Engineering

The details of computing pivot information is presented in Algorithm 4. P is a set of randomly sampled pivots. ϵ_0, δ_0 are input parameter to determine approximation ratio of computation. For each pivot p, we consider it as the query location and compute the seed set by varying k from 1 to $k_{\max}$. For each pair of k, p, we use Algorithm 3 to derive the lower bound of L_p^k and compute the sample size l_p required for selecting the seed set. If current sample size is smaller than l_p, we add more samples in Line 7 and select the seed set in Line 8 based on the Algorithm 2. Finally, we store the estimated influence of the selected seed set into P_{inf} in Line 9.

```
Algorithm 4: Compute Pivot Information
   Input  : $\mathcal{G}$ : a social network, $k_{\max}$ : maximum $k$, $P$ :
            a set of pivots, $\epsilon_0, \delta_0$ : parameters for pivots
            information.
   Output : $P_{inf}$ : pivots information
 1 $P_{inf}, \mathcal{R} \leftarrow \emptyset$ ;
 2 for each $p \in P$ do
 3     for each $k \in [1, k_{\max}]$ do
 4         $L_p^k \leftarrow$ LowerBoundEstimation($k$, $p$) ;
 5         $l_p \leftarrow l(\epsilon_0, \delta_0, k, p, L_q^k)$ ;
 6         if $|\mathcal{R}| < l_p$ then
 7             Add $l_p - |\mathcal{R}|$ samples to $\mathcal{R}$ ;
 8         Select seed set $S_p^k$ with the first $l_p$ samples ;
 9         $P_{inf} \leftarrow P_{inf} \cup (k, p, \hat{I}_p(S_p^k))$ ;
10 return $P_{inf}$
```

4.3.2 Derive Lower Bound

Given the pivots information calculated in Algorithm 4, for a given query q, we select its nearest pivot to derive the lower bound of OPT_q^k according to Lemma 8.

Lemma 8. *Given a pivot p and an integer k, $\epsilon_0, \epsilon_1, \epsilon_2$ and δ_0 are set as in Lemma 7. The sample size is larger than $l(\epsilon_0, \delta_0, p, k, OPT_p^k)$. Then given a query location q, L_q^k in Equation* (13) *is a lower bound of OPT_q^k with at least $1 - \delta_0$ probability.*

$$L_k^q = \frac{1 - 1/e - \epsilon_0}{1 - 1/e - \epsilon_0 + \epsilon_2} \cdot \exp(-\alpha d(p, q)) \cdot \hat{I}_P(S_p^k) \quad (13)$$

Proof: Since the sample size is larger than $l(\epsilon_0, \delta_0, p, k, OPT_p^k)$, which fulfills the requirement in Lemma 6 and 7, then we have the following equations hold with $1 - \delta_0$ probability.

$$I_p(S_p^k) \geq (1 - 1/e - \epsilon_0) OPT_p^k \quad (14)$$

$$\hat{I}_q(S_p^k) - I_q(S_p^k) < \epsilon_2 \cdot OPT_q^k \quad (15)$$

Combine the two equations, we have:

$$I_p(S_p^k) \geq \frac{1 - 1/e - \epsilon_0}{1 - 1/e - \epsilon_0 + \epsilon_2} \cdot \hat{I}_P(S_p^k)$$

In addition, we have:

$$\begin{aligned} OPT_q^k &\geq I_q(S_p^k) \\ &= \sum_{v \in V} I(S_p^k, v) w(v, q) \\ &\geq \sum_{v \in V} I(S_p^k, v) w(v, p) \exp\left(-\alpha d(p, q)\right) \\ &= I_p(S_p^k) \cdot \exp\left(-\alpha d(p, q)\right) \\ &\geq \exp\left(-\alpha d(p, q)\right) \cdot \frac{1 - 1/e - \epsilon_0}{1 - 1/e - \epsilon_0 + \epsilon_2} \cdot \hat{I}_p(S_p^k) \end{aligned}$$

Since Equation (14) and Equation (15) hold with $1 - \delta_0$ probability. Therefore the lemma is correct. □

Based on Lemma 8, after selecting a pivot, we can use the estimated influence $\hat{I}_p(S_p^k)$ to derive the lower bound. The tightness of the lower bound is only decided by the distance between the pivot and the query location when k, ϵ_0, δ_0 is fixed.

4.3.3 Build Index

Given Lemma 8 and the pivots information, we are able to derive the lower bound of OPT_q^k for any query location q. Algorithm 5 describes the details of building the index for the RIS-DA approach. There are two sets of input parameters, for pivot computation and for online DAIM queries respectively. This is because, we hope that the derived lower bound in Lemma 8 should be as tight as possible with high probability, in order to avoid generating to many samples for online processing. In addition, computing the pivots information is offline, in which we can afford more computation time, so we set ϵ_0, δ_0 smaller than ϵ, δ.

Since the tightness of the lower bound is decided by the distance between the query location to its nearest pivot. We can partition the space into Voronoi diagram based on the set of pivots in Line 2. For each Voronoi cell $c(p)$ with pivot p as its corresponding center, we find the furthest location $q_{c(p)}$ from p in $c(p)$ (Line 4). Then we use Lemma 8 to derive the lower bound of $OPT_{q_{c(p)}}^k$. Next, we calculate the sample size required for online query processing in Line 7. Since $q_{c(p)}$ is the furthest location from p in $c(p)$, the corresponding lower bound has the smallest value for all the locations in $c(p)$ and the sample size calculated will be the largest. Therefore, the derived sample size is large enough to answer any query in $c(p)$ with theoretical guarantee. We repeat these process for all the Voronoi cells. Finally, we add enough samples for index.

For Online Query. For an online query q, k, ϵ, δ, we first find its nearest pivot and derive the lower bound L_q^k based on Lemma 8. Then we compute the sample size required, *i.e.*, $l(\epsilon, \delta - \delta_0, q, k, L_q^k)$. Recall that the lower bound holds with at least $1 - \delta_0$ probability. Then we use first $l(\epsilon, \delta - \delta_0, q, k, L_q^k)$ samples and Algorithm 2 to compute the seed set. This is because the indexed samples have considered the worst case scenario, which can be much larger than $l(\epsilon, \delta - \delta_0, q, k, L_q^k)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2633472, IEEE Transactions on Knowledge and Data Engineering

Algorithm 5: Index Construction

```
Input  : $\mathcal{G}$ : a social network, $k_{\max}$ : maximum $k$,
         $\epsilon_0, \delta_0$ : parameters for pivots information,
         $\epsilon, \delta$ : parameters for online DAIM queries,
         $P_{inf}$ : pivot information.
Output : $\mathcal{R}$ : the indexed samples.
$\mathcal{R} \leftarrow \emptyset$; $l_{\max} \leftarrow 0$ ;
Build Voronoi diagram based on $P$ ;
for each $p \in P$ do
    $q_{c(p)} \leftarrow$ Furthest location to $p$ in $c(p)$ ;
    for each $k \in [1, k_{\max}]$ do
        Compute the lower bound $L^k_{q_{c(p)}}$ ;
        $l_{\max} \leftarrow \max(l_{\max}, l(\epsilon, \delta - \delta_0, k, q_{c(p)}, L^k_{q_{c(p)}}))$ ;
Add $l_{\max}$ samples to $\mathcal{R}$ ;
return $\mathcal{R}$
```

Lemma 9. *Given a query location q, an integer k and input parameters $\epsilon_0, \delta_0, \epsilon, \delta$, Algorithm 5 indexes sufficient samples to return a $1 - 1/e - \epsilon$ result with at least $1 - \delta$ probability.*

Proof Sketch: In Algorithm 5, by partitioning the space into Voronoi cells, we derive the lower bound of the optimal influence for all the queries. Thus, the samples indexed are sufficient for answer any query with theoretical guarantee. □

In some scenarios, companies may have many stores, and they want to conduct a multi-store promotion. In Appendix E, we discuss the opportunity and challenges about the multi-location query.

5 Experiments

In this section, we present the results of a comprehensive performance study on real-world geo-social networks to demonstrate the effectiveness and efficiency of the techniques proposed in this paper.

5.1 Experiment Setup

Algorithms. The algorithms demonstrate are listed as follows.

- **PMIA**: we extend the PMIA method in [3], to support the DAIM problem. For PMIA, we pre-compute the $MIIA(u)$ and $MIOA(u)$ for each node.
- **MIA-DA**: the MIA-based approach, which is proposed in Section 3.
- **RIS-DA**: the RIS-based approach, which is presented in Section 4.

Dataset. To evaluate the algorithms, in this paper, we use four real-world geo-social networks where users can share their check-ins. These checkins serve as users' locations and are widely used in many researches about geo-social networks [8], [9]. The data information are shown in Table 2. Brightkite and Gowalla datasets are obtained from https://snap.stanford.edu/data, and the other two datasets are obtained from [8]. For Brightkite and Gowalla, 88.6% and 54.4% users have checkins respectively. For the other two datasets, each user has a checkin. For users who has multiple checkins, we randomly select one for him. For users who do not have checkin, we randomly generate one from the whole space.

Dataset	# of Nodes (n)	# of Edges (m)
Brightkite	58K	428K
Gowalla	197K	1.9M
Twitter	554K	4.29M
Foursqure	4.9M	53.7M

TABLE 2
Experiment Datasets

Propagation Probability. We use the weighted cascade model to generate the probability over each edge. The probability of edge $\langle u, v \rangle$ is set as $\frac{1}{|N_{in}(v)|}$, where $|N_{in}(v)|$ is the number of incoming neighbours of v.

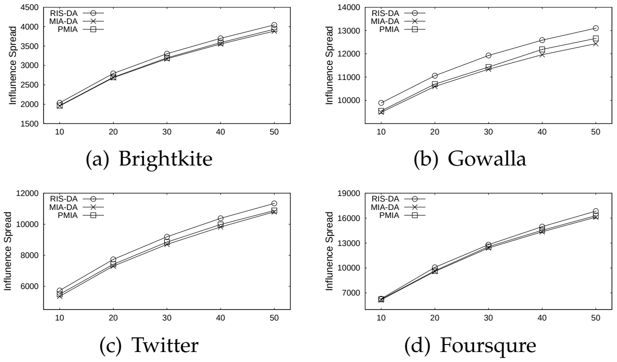
Workload and Parameters. In this paper, we focus on efficiently processing each query based on the index. To evaluate the efficiency, the average response time for all the queries is reported. To evaluate the effectiveness, we report the average influence spread of the returned seed set. The query locations are randomly selected from the entire space. The seed set size k varies from 10 to 50 with 30 as the default value. For the node weight function, c is set to 1, and α varies form 0.001 to 0.01 with 0.01 as default value. For the MIA-DA approach, the anchor point set size $|\mathcal{L}|$ is set to 300, τ is set to 200 for the region based estimation, and the threshold θ is set to 0.05. For the RIS-DA approach, the number of sampled pivots is set to 2000. For the parameters of computing pivot, $\epsilon_0 = 0.1$ and $\delta_0 = 1/10n$. For the parameters for online processing, $\epsilon = 0.5$ and $\delta = 1/n$.

To obtain the influence spread, we run 10000 round random simulations for each returned seed set and report the average of the influence spread. All algorithms are implemented in C++ with GNU GCC 4.8.2 with -O3 flag. Experiments are conducted on a PC with Intel Xeon 3.4GHz CPU and 96G memory using Redhat Linux.

5.2 Effectiveness Evaluation

To evaluate the effectiveness of the proposed algorithms, we compare them with PMIA on the four datasets, the average influence spread is reported. The results are shown in Figure 3 by varying k.

Summary of Results. *1)* MIA-DA obtains slightly smaller influence spread compared with PMIA. *2)* RIS-DA returns the largest influence spread among the three evaluated methods.



(a) Brightkite

(b) Gowalla

(c) Twitter

(d) Foursqure

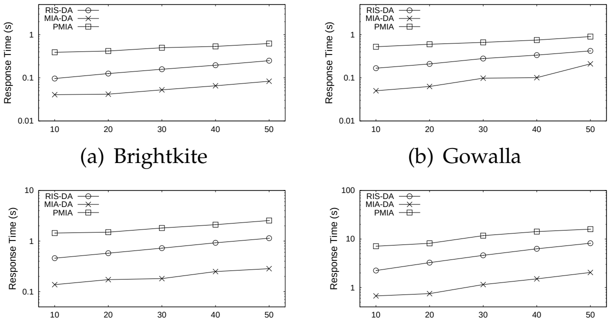
Fig. 3. Effectiveness Evaluation

From Figure 3, we can see, with the increase of k, the influence spread increases on all the datasets. For most cases, the influence spread returned by MIA-DA and PMIA are almost the same. PMIA can return a slightly larger influence spread than MIA-DA on a few cases, this is because PMIA considers the influence propagation through alternative pathes, instead of only considering the maximum influence path as in MIA-DA. RIS-DA always returns the largest influence spread compared with the other two methods, because RIS-DA has bounded theoretical guarantee.

5.3 Efficiency Evaluation

To evaluate the efficiency of the proposed algorithms, we report the response time by comparing with PMIA. Figure 4 reports the efficiency of the algorithms by varying k on four datasets.

Summary of Results. MIA-DA runs fastest among all the algorithm, and RIS-DA outperforms PMIA in efficiency.



(a) Brightkite

(b) Gowalla

(c) Twitter

(d) Foursqure

Fig. 4. Efficiency Evaluation

As shown in Figure 4, the algorithms proposed in this paper always outperform PMIA, especially MIA-DA which speedups the query processing up to an order of magnitude. For PMIA, even though $MIIA(u)$ and $MIOA(u)$ are pre-computed for each node $u \in V$, it is still necessary to scan the index structure to compute the influence spread and marginal influence for each node, since the weight of nodes is not known in advance. While for MIA-DA, based on the pruning rules, we can efficiently calculate the upper and lower bound of each node's influence spread. By using the priority-based search algorithm, many nodes will not be evaluated and can be pruned directly. RIS-DA runs faster than PMIA, because the samples needed are offline indexed. In addition, we on the fly compute the sample size needed for the given query instead of using all the samples, since build the bipartite graph and compute each initial weighted coverage takes the majority of computation cost. Based on the performance study, the MIA-DA method is faster than the RIS-DA method, but it returns smaller influence spread. In hence, users can make a trade-off between the efficiency and effectiveness.

5.4 Parameter Evaluation

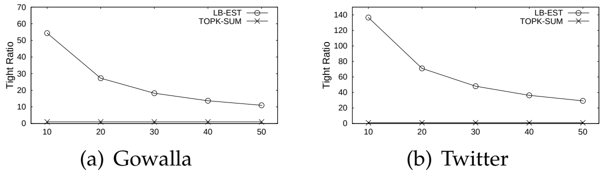
In this section, we study the effects of different parameters and settings for the proposed methods on Gowalla and Twitter datasets.

5.4.1 Lower Bound for Pivots in RIS-DA

In the offline index building phase, we utilize a two-hop neighbours based heuristic method (Algorithm 3) to derive the influence lower bound for a given pivot, instead of directly using the weight sum of the top-k weighted nodes, denoted as TOPK-SUM. The method in Algorithm 3 is denoted as LB-EST. To measure the effectiveness of LB-EST, we define the tightness ratio for a method X as follows.

$$\text{tightness\_ratio(X)} = \frac{\text{lower bound obtained by X}}{\text{lower bound obtained by TOPK-SUM}}$$

Then we have tightness_ratio(TOPK-SUM) = 1, and larger value means tighter lower bound. Figure 5 shows the tightness ratio of the two methods by varying k from 10 to 50. As we can see, LB-EST consistently provides a tighter lower bound than TOPK-SUM. Since the sample size is proportional to the reverse of lower bound, LB-EST can greatly reduce the number of sample required.



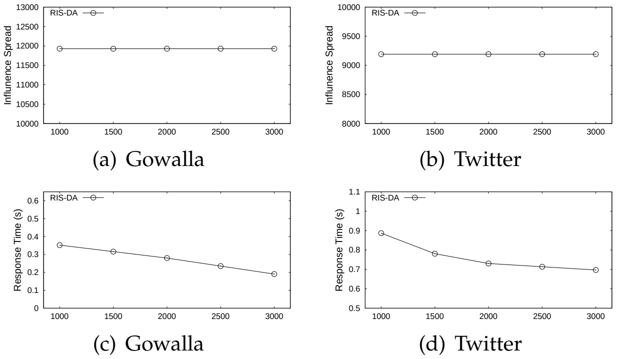
(a) Gowalla

(b) Twitter

Fig. 5. Lower Bound for Pivots in RIS-DA

5.4.2 Number of Pivots in RIS-DA

In this subsection, we evaluate the effects of the number of pivots in RIS-DA, and the experiments results are shown in Figure 6 by varying the number of pivots from 1000 to 3000. As shown in Figure 6(c) and 6(d), the response time decreases when the number of pivots increases. This is because the expected distance between the query location and the selected pivot decreases when the number of pivots increases, and we can derive a tight sample size. While the influence spread of the returned results barely changes due to the same error guarantee.



(a) Gowalla

(b) Twitter

(c) Gowalla

(d) Twitter

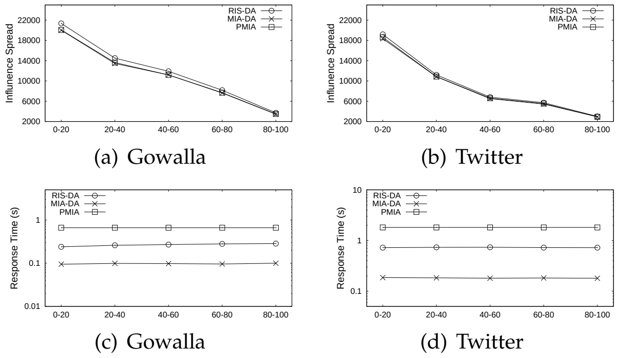
Fig. 6. Number of Pivots in RIS-DA

5.4.3 Average Distance Between Users and Query

As shown in Figure 7, we measure the effects of the average distance between users and the query location. Specially, we partition the set of queries into 5 partitions based on the average distance to the query location. 0-20 denotes the set of queries with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2016.2633472, IEEE Transactions on Knowledge and Data Engineering

smallest average distance, while 80-100 denotes the partition with the largest average distance. As we can see, when the average distance increases, the influence spread decreases, since the average weight of users decreases. While the processing time just slightly changes, since the bounds used in the proposed methods mainly depend on the distance between the query location and the selected sample location.



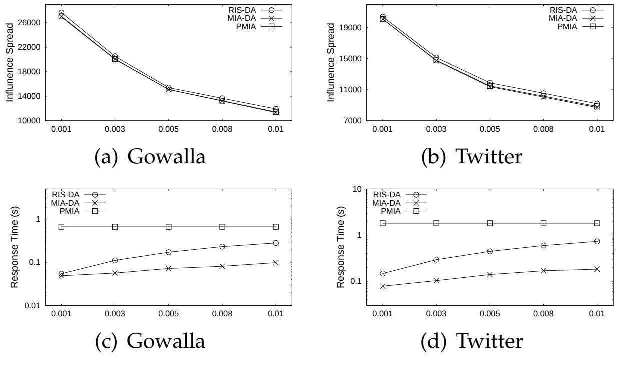
(a) Gowalla (b) Twitter

(c) Gowalla (d) Twitter

Fig. 7. Average Distance Between Users and Query

5.4.4 Parameter α

Since the tightness of the bounds in both MIA-DA and RIS-DA are decided by the parameter α, we report the performance by varying α from 0.001 to 0.01. The experiment results are shown in Figure 8. As we can see in Figure 8(a) and 8(b) with the increase of α, the influence spread decreases, since the weight of each node is reverse proportional to α. The processing time of MIA-DA and RIS-DA both increases when α increases. This is because α denotes the weight decay speed from the query location. Extremely, when $\alpha = 0$, each node has the same weight. Then the seed set calculated at any location are the same. So smaller α will lead to tighter bound for both MIA-DA and RIS-DA, and reduce the searching time.



(a) Gowalla (b) Twitter

(c) Gowalla (d) Twitter

Fig. 8. Impact of Parameter α

6 Related Work

Influence Maximization in Social Network. There is a large amount of literature on the influence maximization problem [1], [2], [3], [4], [15], [19], [20], [21], [22], [23], [24]. Kempe et al. [4] formally define independent cascade model and linear threshold model, and prove the submodulor and monotonic property of the influence spread function and hardness of the problem. In addition, authors present a solution with a $1 - 1/e - \epsilon$ approximation ratio using the greedy framework.

To improve efficiency and maintain the same approximation ratio, Leskovec et al. [21] authors propose a lazy forward framework and achieve 700 times speedup compared to the naive greedy algorithm in [4]. Goyal et al. [24] further ameliorate Leskovec et al.'s methods with a 50% improvement in query time. Recently, Borgs et al. [15] propose a near-linear time approach **RIS** by sampling Random Reverse Reachable Set. Based on the RIS model, Tang et al. [1], [2] further improve its efficiency in term of sample complexity and apply it to a more general diffusion model. Lucier et al. [25] propose a progressive sampling approach which can be applied to parallel framework to calculate the influence spread. In [3], the authors prove that it is #P-hard to calculate the influence spread. Thus, there are many algorithms rely on heuristic strategies to enhance performance. Chen et al. [22] utilized a degree discount heuristic to identify influential nodes. Chen et al. [3] propose the **PMIA** approach, in which the influence is considered to prorogate only through the maximum influence path between users. A similar idea is also applied in [23] to solve the problem under linear threshold model. In [5], Cohen et al. propose a bottom-k sketch based approach to reduce the cost influence estimation. The materialized sketch can be used as an oracle to evaluate the influence of any subset users. Unlike existing works where each user is equally treated, we emphasize the differences between users in a geo-social network when the promoted locations are varied. In addition, we are intent on meeting the online query requirement by constructing the index offline. In [11], the authors also consider a weighted influence maximization model, where the weight for each user is decided by the input keywords and the keywords of user's. However, their index strategy cannot be trivially extended to support our problem.

Influence Maximization in Geo-social Network. With the advance of location enabled devices, the geo-factor plays an increasingly important role in social network analysis. Zhang et al. [26] attempt to measure the influence between users by considering both social relation and location information, and aim to identify influential events. Zhu et al. [9] consider a geo-social network in which each user is associated with multiple check-ins. Given a promoted location, it aims to learn the influence between users based on their check-in distribution. The work most related to ours is by Li et al. [8], who attempt to find a seed set that will maximize the influence spread in a query region. As stated in Section 1, it is non-trivial to determine an appropriate query range when conducting a location aware promotion.

7 Conclusion

Influence maximization is a key problems in viral marketing, given a budget k, in which the aim is to identify a set of users in the social network to maximize the expected influence over all the users. With the proliferation of position enabled devices, many real world applications require the location-aware product promotion. In this paper, we investigate the problem of distance-aware influence maximization. In our previous conference paper [10], we formally define the problem and propose an index-based approaches, MIA-DA, which extends the MIA model for influence calculation. Since the MIA-DA approach is a heuristic method, to improve the effectiveness, in this journal extension, we develop the RIS-DA approach, which extends the RIS model to support the DAIM query. In RIS-DA, we carefully analyze the sample size needed for indexing. Based on the indexed samples, for a given query, RIS-DA is able to return a $1 - 1/e - \epsilon$ result with at least $1 - \delta$ probability. Lastly, we demonstrate the efficiency and effectiveness of proposed techniques on four real geo-social networks.

REFERENCES

[1] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *SIGMOD*, 2015, pp. 1539–1554.

[2] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: near-optimal time complexity meets practical efficiency," in *SIGMOD*, 2014, pp. 75–86.

[3] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *SIGKDD*, 2010, pp. 1029–1038.

[4] D. Kempe, J. M. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *SIGKDD*, 2003, pp. 137–146.

[5] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *CIKM*, 2014, pp. 629–638.

[6] I. R. Misner, "The worlds best known marketing secret: Building your business with word-of-mouth marketing," in *Bard Press, 2nd edition*, 1999.

[7] J. Nail, "The consumer advertising backlash," in *Forrester Research and Intelliseek Market Research Report*, 2004.

[8] G. Li, S. Chen, J. Feng, K. Tan, and W. Li, "Efficient location-aware influence maximization," in *SIGMOD 2014*, 2014, pp. 87–98.

[9] W. Zhu, W. Peng, L. Chen, K. Zheng, and X. Zhou, "Modeling user mobility for location promotion in location-based social networks," in *SIGKDD*, 2015, pp. 1573–1582.

[10] X. Wang, Y. Zhang, W. Zhang, and X. Lin, "Distance-aware influence maximization in geo-social network," in *ICDE*, 2016, pp. 1–12.

[11] Y. Li, D. Zhang, and K. Tan, "Real-time targeted influence maximization for online advertisements," *PVLDB*, 2015.

[12] S. Chen, J. Fan, G. Li, J. Feng, K. Tan, and J. Tang, "Online topic-aware influence maximization," *PVLDB*, 2015.

[13] C. Y. Wei Chen, Tian Lin, "Real-time topic-aware influence maximization using preprocessing," in *Arxiv.org*, 2014.

[14] Ç. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates, "Online topic-aware influence maximization queries," in *EDBT*, 2014, pp. 295–306.

[15] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *SODA*, 2014, pp. 946–957.

[16] Y. Wu, S. Yang, and X. Yan, "Ontology-based subgraph querying," in *ICDE*, 2013, pp. 697–708.

[17] K. Feng, G. Cong, S. S. Bhowmick, and S. Ma, "In search of influential event organizers in online social networks," in *SIGMOD*, 2014, pp. 63–74.

[18] C. Borgs, M. Brautbar, J. T. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *SODA*, 2014, pp. 946–957.

[19] P. Domingos and M. Richardson, "Mining the network value of customers," in *KDD*, 2001, pp. 57–66.

[20] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *KDD*, 2002, pp. 61–70.

[21] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *KDD*, 2007, pp. 420–429.

[22] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *KDD*, 2009, pp. 199–208.

[23] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *ICDM*, 2010, pp. 88–97.

[24] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," in *WWW*, 2011, pp. 47–48.

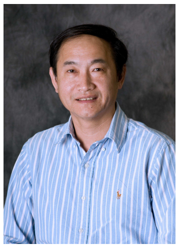
[25] Y. S. Brendan Lucier, Joel Oren, "Influence at scale: Distributed computation of complex contagion in networks," in *KDD*, 2015.

[26] C. Zhang, L. Shou, K. Chen, G. Chen, and Y. Bei, "Evaluating geo-social influence in location-based social networks," in *CIKM*, 2012.

Xiaoyang Wang is a research associate in the University of Technology, Sydney. He received his BSc and MSc degrees in Computer Science from Northeastern University, China in 2010 and 2012 respectively, and PhD degree from the University of New South Wales, Australia in 2016. His research interests include query processing on massive spaital data stream.

Ying Zhang is a senior lectuer and ARC DECRA research fellow (2014-2016) at QCIS, the University of Technology, Sydney (UTS). He received his BSc and MSc degrees in Computer Science from Peking University, and PhD in Computer Science from the University of New South Wales. His research interests include query processing on data stream, uncertain data and graphs. He was an Australian Research Council Australian Postdoctoral Fellowship (ARC APD) holder (2010-2013).

Wenjie Zhang is currently a senior lecturer and ARC DECRA (Australian Research Council Discovery Early Career Researcher Award) Fellow in School of Computer Science and Engineering, the University of New South Wales, Australia. She received PhD in computer science and engineering in 2010 from the University of New South Wales. Since 2008, she has published more than 20 papers in SIGMOD, VLDB, ICDE, TODS, TKDE and VLDBJ. She is the recipient of Best (Student) Paper Award of National DataBase Conference of China 2006, APWebWAIM 2009, Australasian Database Conference 2010 and DASFAA 2012, and also co-authored one of the best papers in ICDE2010, ICDE 2012 and DASFAA 2012. In 2011, she received the Australian Research Council Discovery Early Career Researcher Award.



Xuemin Lin is a Scientia Professor in the School of Computer Science and Engineering, the University of New South Wales. He has been the head of database research group at UNSW since 2002. He is a concurrent professor in the School of Software, East China Normal University. Before joining UNSW, Xuemin held various academic positions at the University of Queensland and the University of Western Australia. Dr. Lin got his PhD in Computer Science from the University of Queensland in 1992 and his BSc in Applied Math from Fudan University in 1984. During 1984-1988, he studied for Ph.D. in Applied Math at Fudan University. He currently is an associate editor of ACM Transactions on Database Systems. He is a senior member of IEEE. His current research interests lie in data streams, approximate query processing, spatial data analysis, and graph visualization.