Distance-Aware Influence Maximization in Geo-social Network

Xiaoyang Wang and Ying Zhang
QCIS, University of Technology Sydney, Australia
{Xiaoyang.Wang, Ying.Zhang}@uts.edu.au

Wenjie Zhang and Xuemin Lin
University of New South Wales
{zhangw, lxue}@cse.unsw.edu.au

***Abstract*—Influence maximization is a key problem in viral marketing. Given a social network G and a positive integer k, it aims to identify a seed set of k nodes in G that can maximize the expected influence spread in a certain propagation model. With the proliferation of geo-social networks, location-aware product promotion is becoming more necessary in real applications. However, the importance of the distance between users and the promoted location is underestimated in existing models. For instance, when opening a restaurant in downtown, through online promotion, the owner may expect to influence more customers who are close to the restaurant, instead of people that are far away from it. In this paper, we formally define the distance-aware influence maximization problem, to find a seed set that maximizes the expected influence over users who are more likely to be the potential customers of the promoted location. To efficiently calculate the influence spread, we adopt the maximum influence arborescence (MIA) model for influence approximation. To speed up the search, we propose three pruning strategies to prune unpromising nodes from expensive evaluation and achieve potential early termination in each iteration without sacrificing the final result's approximation ratio. In addition, novel index structures are developed to compute the bounds used in the three pruning strategies. By integrating these pruning strategies, we propose a priority based algorithm which searches users based on their order of influence. The algorithm achieves an approximation ratio of $1-\frac{1}{e}$ under the MIA model. In the final, comprehensive experiments over real datasets demonstrate the efficiency and effectiveness of the proposed algorithms and pruning strategies.**

I. 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 [18], [19], [5], [11], [8]. Given a social network G and a positive integer k, the influence maximization problem aims to identify k nodes in G, called a seed set, which maximizes the expected number of nodes that are influenced under a certain propagation model. The propagation of influence is based on the trust between families, close friends, co-workers, etc. This marketing strategy is shown to be more effective than traditional advertising channels, such as TV and newspapers[15], [16]. The influence maximization problem has been formally defined by Kemple et al [11], in which two models, i.e., the independent cascade (IC) model and the linear threshold (LT) model, have been used to describe the propagation of influence over the social network. Authors have proved that the problem is NP-hard under both IC and LT models, and have proposed a greedy algorithm with a $1-\frac{1}{e}$ approximation ratio based on the monotonic and submodular property of the influence spread functions.

Motivation. In most existing works concerning the influence maximization problem [18], [19], users in a social network are equally treated. The advance of geo-position enabled devices and services makes it possible to add a spatial dimension to traditional social network, *i.e.,* geo-social network. When conducting a location aware promotion (*e.g.,* promoting a newly opened restaurant in downtown), existing influence maximization model may not fulfil the requirement because of the ignorance of users' spatial information. Intuitively, users who are close to a promoted location are more likely to attend the promoted location. Hence, it is natural to consider that users should be weighted differently with respect to the promoted location. To the best of our knowledge, only two papers [13], [23] have taken location information into consideration when selecting the influential seed set. In [23], 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 [13], 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.

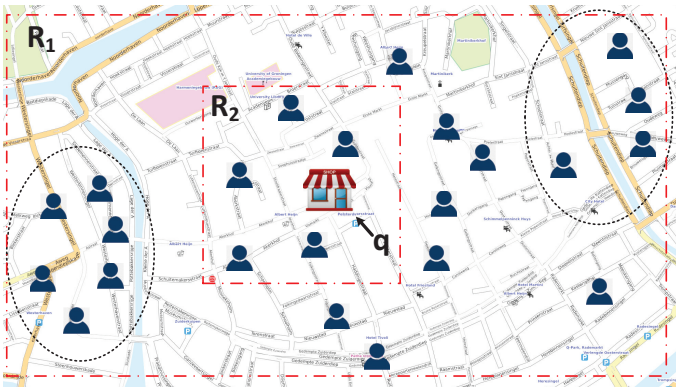


Fig. 1. Motivation Example

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 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 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 [13]. 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 location aware influence maximization. In this paper, we propose the *Distance-Aware Influence Maximization* (DAIM) model which seamlessly combines two factors: influence spread and users' distance from the query location. Given a geo-social network G and a query location q in 2-dimensional space, *i.e.,* 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 user will be. In the DAIM model, the influence gain of activating a user is defined as the activated probability times 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. The first is that the number of nodes to be evaluated is large. Unlike the problem in [13], 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 [5], so it is critical to reduce the number of users to be evaluated. The second one is that it is not an easy task to meet the online requirement. It is important for a system to support the online requirement, since there may be plenty of DAIM queries issued. However, although many works have investigated the influence maximization problem, few of them can meet the online requirement. While some works [13], [14], [4], [20], [1] have attempted to provide algorithms for online service, the problem settings are different from ours, and it is non-trivial to extend these algorithms to solve the problem proposed in this paper.

To address the above challenges, in this paper, we extend the maximum influence arborescence (MIA) model for the purpose of approximating the influence spread. To reduce the number of nodes to be exactly evaluated, we develop three pruning strategies. The first two pruning strategies are based on the estimated bounds of user's influence and marginal influence, which enable us to discard insignificant nodes without calculating their influence or marginal influence. The third rule is based on the case that we can obtain a seed set S' with a bounded approximation ratio $\beta(1-\frac{1}{e})$. We can then achieve possible early termination in each iteration if we find a seed set S which influence is no smaller than $1/\beta$ times that of S'. Even if we early terminate in iterations, the returned seed set S still retains the $1-\frac{1}{e}$ approximation ratio. In addition, novel index structures are proposed to obtain the bounds and seed set required in the three pruning rules. Lastly, by integrating the pruning rules proposed, we develop a priority based algorithm, in which we search the nodes based on their influence order to boost the pruning power of the three rules. Under the MIA model, the algorithm can achieve a seed set of k nodes with $1-\frac{1}{e}$ approximation ratio.

Contributions. Our principle contributions are summarized as follows.

- We formally introduce the distance-aware influence maximization problem over geo-social network and extend the MIA model to support the new problem with an approximation ratio guarantee.
- To prune insignificant nodes from the exact calculation, we develop three pruning rules. Novel index structures and algorithms are proposed to obtain the information needed in the three pruning rules.
- We integrate the three pruning rules and propose a priority based algorithm which searches the nodes based on their order of influence.
- We evaluate the performance of the proposed algorithms and pruning rules on three real geo-social networks. Our comprehensive experiments confirm the efficiency and effectiveness of the proposed techniques.

Roadmap. The rest of the paper is organized as follows. Section II formally introduces the problem and the related techniques used in the paper. Section III introduces three pruning rules and novel index structures for obtaining the information needed for the three rules. In Section IV, we propose the priority based algorithm by integrating the proposed pruning rules. We demonstrate the efficiency and effectiveness of the proposed techniques on real datasets in Section V. Lastly, we introduce the related works in Section VI and conclude the paper in Section VII.

II. Preliminary

We first formally introduce the problem of distance-aware influence maximization in Section II-A. In Section II-B, we present the influence calculation model used in this paper. Table I summarizes the notations frequently used throughout the paper.

A. Problem Definition

We consider a geo-social network as a directed graph $G=(V,E)$, where V represents the set of nodes (users) and E represents the set of edges (relationships between users) in 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 $f: V \times q \rightarrow R$ 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

TABLE I. THE SUMMARY OF NOTATIONS

Notation	Meaning
$G = (V, E)$	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
$f(v, q)$	the weight of node v according to q
$I(S)$	influence of set S
$I(S, v)$	probability of set S influence v
$I_q(S)$	distance-aware influence spread of S
$d(v, q)$	Euclidean distance between v and q
$MIP(u, v)$	maximal influence path between u and v
θ	threshold for pruning insignificant path
$I(u\|S)$	marginal influence of u in IM model
$I_q(u\|S)$	marginal influence of u in DAIM model
$I_q^L(\{u\})$, $I_q^U(\{u\})$	lower and upper bound of $I_q(\{u\})$
$I_q^U(u\|S)$	upper bound of $I_q(u\|S)$

focus on the *independent cascade* (IC) model, which is most widely adopted by existing researches [4], [14], [2], [8]. Under the IC model, each edge $\langle u, v\rangle \in E(G)$ has an independent probability $\mathcal{P}(u, v) \in [0, 1]$, 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 $\mathcal{P}(u, v)$.
- Once a node becomes active, it remains activate for subsequence 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.,* $I(S) = E[\sum_{i=0}^{t} S_i]$. 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 $G = (V, E)$ 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 $\sum_{v \in V} I(S, v) f(v, q)$, where $f(v, q)$ is the weight of v with respect to q.

We refer to distance-aware influence spread as DA influence spread hereafter. As discussed in Section I, users who are close to promoted location should have higher priority to be influenced. Thus $f(v, q)$ is inversely proportional to the distance between v and q. For ease of exposition, in this paper, we only investigate the case where $f(v, q) = ce^{-\alpha d(v,q)}$, which is a widely used decay function [21]. In the function, $c > 0$ is the maximum weight that a node can achieve, $\alpha > 0$ denotes the weight decay speed, 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 G, a query location q and an integer k, the problem of *distance-aware influence maximization* (DAIM) is to find a set S^* of k nodes in G which has the largest distance-aware influence spread, *i.e.,* $S^* = \arg\max_{S \subseteq V}\{I_q(S) || S| = k\}$. S^* is called a seed set and each node $u \in S^*$ is called a seed.

Problem Hardness. We show the hardness of the problem by considering a simple case with $c = 1$ and $\alpha = 0$ in function f. In this case, each node's weight equals 1, thus the DAIM problem becomes a traditional influence maximization problem which is NP-hard [11]. Therefore, *the DAIM problem is an NP-hard problem*. Since calculating the influence spread of a set S of nodes is #P-hard [5], following a similar routine, *the computation of DA influence spread $I_q(S)$ is also a #P-hard problem.*

B. MIA Model based Approximation

To solve the DAIM problem, a fundamental step is to calculate the DA influence spread for a set of nodes, which is shown to be #P-hard. Many heuristic approaches have been proposed in the literature to enhance the performance. In this paper, we extend the MIA model [5] to approximate the influence propagation and influence spread calculation for the DAIM problem. However, the techniques developed can also be extended to support other models. Hereafter, when there is no ambiguity, $I_q(S)$ and $I(S)$ denote the influence spread of S under the MIA model.

Influence Approximation. Given a geo-social network 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 $\mathcal{P}(p(u, v)) = \prod_{i=1}^{m-1} \mathcal{P}(w_i, w_{i+1})$. 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.,*

$$MIP(u, v) = \underset{p \in p(u,v,G)}{\arg\max} \{\mathcal{P}(p)\} \quad (1)$$

where $p(u, v, G)$ denotes all the paths between u and v in 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. Note particularly that if $\mathcal{P}(MIP(u, v)) < \theta$, 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 or we say that $\mathcal{P}(MIP(u, v)) = 0$. The distance aware influence from u to v denoted as $I_q(\{u\}, v)$, is then calculated as $\mathcal{P}(MIP(u, v)) f(v, q)$. If we translate each edge's probability $\mathcal{P}(u, v)$ with $-\log(\mathcal{P}(u, v))$, computing $MIP(u, v)$ is equal to identifying the shortest path from u to v with distance smaller than $-\log(\theta)$. In the examples that follow, we use $\theta = 0.25$.

Given a set S of nodes, $I_q(S, v)$ does not equal $\sum_{u \in S} I_q(u, v)$, since there is correlation between the nodes in S. For example, in Figure 2, we have a geo-social network. Given a query q, the weight of each node is listed on the right. $I_q(\{v_1\}, v_5) = 0.5$ and $I_q(\{v_6\}, v_5) = 0.9$, while

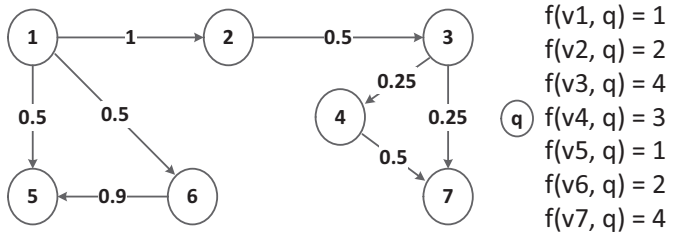


Fig. 2. Example of DA Influence Spread

$I_q(\{v_1, v_6\}, v_5) = (1 - (1 - 0.5)(1 - 0.9)) \times 1 = 0.95 < 1.4$. To calculate the probability that a given seed set S influences v, we utilize the *Maximum Influence In(Out) Arborescence* structure [5], which assembles the maximal influence paths to(from) a node v.

Definition 2 (Maximum Influence In(Out) Arborescence): Given a geo-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 G. To influence v, S must firstly influence the neighbours $N(v)$ of v in $MIIA(v)$. Note that $N(v)$ does not equal the incoming neighbours of v in G. If $v \in S$, then $I_q(S, v) = f(v, q)$. By traversing $MIIA(v)$ from v to the leaf nodes, we can calculate $I_q(S, v)$ as shown in Equation (2).

$$I_q(S, v) = (1 - \prod_{w \in N(v)} (1 - \mathcal{P}(S, w, v)\mathcal{P}(w, v)))f(v, q) \quad (2)$$

where $\mathcal{P}(S, w, v)$ is the probability that S will influence w through $MIIA(v)$ and $\mathcal{P}(S, w, v) = 1$ when $w \in S$. Given a query location q, to calculate the DA influence spread of S under the MIA model, we simply sum the influence of S to every node in G, *i.e.,*

$$I_q(S) = \sum_{v \in V} I_q(S, v) \quad (3)$$

and the marginal influence $I_q(u|S)$ for $u \notin S$ is equal to $I_q(S \cup \{u\}) - I_q(S)$.

Complexity Analysis. After relaxing the influence calculation with the MIA model, we can compute $I_q(S)$ efficiently. However, the problem of finding a set of k nodes that will maximize the DA influence spread is still NP-hard, as stated in Lemma 1.

Lemma 1: Under the MIA model, the problem of computing DAIM is NP-hard.

Proof: To prove the Lemma, we consider a simple case by setting $\alpha = 0$ and $c = 1$. The DAIM problem then equals the influence maximization problem for any query location under the MIA model in [5] which has been proved to be NP-hard. Consequently, the DAIM problem is NP-hard under the MIA model. ■

Greedy Approach. Even though the DAIM problem is NP-hard, the DA influence spread function has two properties: 1) $I_q(S)$ is monotonic, *i.e.,* if two sets S, $T \subseteq V$ and $S \subseteq T$, we have $I_q(S) \leq I_q(T)$. 2) $I_q(S)$ is submodular, *i.e.,* if two sets S, $T \subseteq V$, $S \subseteq T$ and $v \in V \setminus S \cup T$, we have $I_q(T \cup \{v\}) - I_q(T) \leq I_q(S \cup \{v\}) - I_q(S)$. The correctness of the two properties can be reduced from the proof in [5]. Therefore, we can adapt the greedy framework in Algorithm 1 to obtain a set of nodes with approximation ratio $1 - \frac{1}{e}$ under the MIA model.

Algorithm 1: Greedy Algorithm

Input : G : a geo-social network; k : seed set size; q : query location.
Output : S : a set of k nodes
1 Off-line compute $MIIA(v)$ and $MIOA(v)$ for $v \in V$;
2 $S \leftarrow \emptyset$;
3 **for** $i = 1$ to k **do**
4 $\quad u \leftarrow \arg\max_{w \in V \setminus S}(I_q(S \cup \{w\}) - I_q(S))$;
5 $\quad S \leftarrow S \cup \{u\}$;
6 **return** S

Since there many queries are raised, the $MIIA(v)$ and $MIOA(v)$ structure are used repeatedly. In Algorithm 1, we pre-compute the $MIIA(v)$ and $MIOA(v)$ off-line for each node $v \in V$ in Line 1. For each iteration from Lines 3 to 5, we select the node u with maximal marginal influence $I_q(u|S)$ under the MIA model. Chen et al. [5] have provided an efficient approach for calculating the marginal influence for all nodes.

III. Bounded Influence

The main limitation of Algorithm 1 are that it is required to compute the influence or marginal influence for many nodes. However, most of these nodes are insignificant, *i.e.,* they have small influence. Thus we should avoid conducting the exact evaluation of these nodes. Suppose we have an oracle which allows us to achieve the upper and lower bounds of each node's influence, and the upper bound 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 $I_q(\{u\}) \geq I_q^U(\{v\})$ or $I_q^L(\{u\}) \geq I_q^U(\{v\})$, we can directly prune v from the first seed selection.

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

where S denotes the currently selected seed set, and u and v are nodes which are not in S, $I_q^U(\{u\})$, $I_q^L(\{u\})$ and $I_q^U(u|S)$ denote the upper bound of $I_q(\{u\})$, lower bound of $I_q(\{u\})$ and upper bound of $I_q(u|S)$ respectively.

Occasionally, we are permitted to "relax" the criteria of returned seed set, that is we only have to return a seed set with at least $1 - \frac{1}{e}$ approximation ratio, instead of the seed set computed exactly following the greedy framework. Assuming we have obtained a seed set T with $\beta \times (1 - \frac{1}{e})$ approximation ratio, *i.e.,* $I_q(T) \geq \beta(1 - \frac{1}{e})I_q(S^*)$, where $\beta \in (0, 1]$ and S^* is the seed set with maximal influence spread, we can

propose the following early termination rules. More details about the motivation and definition of Rule 3 will be presented in Section III-C.

Rule 3. If we have found a set set S' with $I_q(S') \geq I_q(T)/\beta$, we can terminate and return S'.

Based on Rules 1 and 2, we can prune insignificant nodes from evaluation, and we can realize possible early termination through Rule 3 without harming the approximation ratio guarantee. In this section, we investigate the techniques for obtaining the bounds used in the above three rules by using a pre-computed index. We introduce the approaches to develop the lower and upper bounds of $I_q(\{u\})$ in Section III-A. In Section III-B, we introduce the method of obtaining the upper bound of $I_q(u|S)$. Lastly, in Section III-C, we propose the solution for obtaining the information needed in Rule 3. The main algorithm for selecting the seed set using the above three rules will be introduced in Section IV. Hereafter, we assume that $MIIA(u)$ and $MIOA(u)$ are all computed off-line.

A. Rule 1: Influence Bound Estimation

In this section, we develop several approaches to derive the influence bounds used in Rule 1.

Anchor Point based Estimation. If we are able to enumerate all the query locations, we can answer any query immediately by pre-computing the solution off-line. Unfortunately, there are infinite query locations in space. However, we can pre-compute the influence of nodes in respect of some sample locations. By investigating the relationship between the query location and the sample locations, we are able to approximate the influence of each node on-the-fly.

Anchor Points. Formally, an anchor point is a point in 2-dimensional space. $\mathcal{L}$ denotes a set of anchor points selected off-line. For each anchor point $a_i \in \mathcal{L}$, we compute the distance aware influence $I_{a_i}(\{u\})$ of each node u in V under the MIA model by taking a_i as the query location.

Derive Bounds. Based on Definition 1, it is easy to verify that Lemma 2 is correct.

Lemma 2: Given different query locations, the probability that S influences any node v is unchanged.

According to Lemma 2, the reason that $I_{q_i}(S, v) \neq I_{q_j}(S, v)$ in DAIM problem is because the weight of v may be different given two query locations q_i and q_j. Given a query location q and an anchor point a_i, based on triangular inequality, for $v \in V$ we have:

$$d(v, a_i) - d(q, a_i) \leq d(v, q) \leq d(q, a_i) + d(v, a_i)$$

In consequence, we have:

$$\begin{aligned} I_q(S) &= \sum_{v \in V} I(S, v) c \exp(-\alpha d(v, q)) \\ &\leq \sum_{v \in V} I(S, v) c \exp(-\alpha (d(v, a_i) - d(a_i, q))) \\ &= \sum_{v \in V} I_{a_i}(S, v) \exp(\alpha d(a_i, q)) \\ &= I_{a_i}(S) \exp(\alpha d(a_i, q)) \end{aligned}$$

When $S = \{u\}$, we can compute the upper bound of u's influence as follows:

$$I_q^U(\{u\}) = I_{a_i}(\{u\}) \exp(\alpha d(a_i, q)) \quad (4)$$

Following a similar routine, we can compute the lower bound of u's influence as shown in Equation 5.

$$I_q^L(\{u\}) = I_{a_i}(\{u\}) \exp(-\alpha d(a_i, q)) \quad (5)$$

Example 2: As shown in Figure 3, we partition the space into 4 rectangles and select an anchor point in each partition. For any query location there is an anchor point within $d_{min} = 3$, which equals half of the length of partition diagonal line. Given a query location q, we select the nearest anchor point a_1 to estimate the bounds and $d(a_1, q) = 0.5$. Off-line, we compute the user's influence for each anchor point, *e.g.,* $I_{a_1}(\{v_1\}) = 4.75$. For query location q, $I_1(\{v_1\})$ can be bounded by $[\frac{4.75}{e}, 4.75e]$, if $\alpha = 2$ in the decay function.

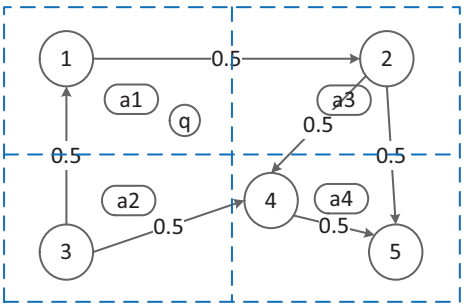


Fig. 3. Example for Pruning Rules

Anchor Points Selection and Maintenance. Based on Equations (4) and (5), the tightness of the estimated bounds is decided by the distance between the query location and the anchor point. Therefore, given a query location, we select the nearest anchor point in $\mathcal{L}$ to generate the influence bounds of each node. Given a space budget $\mathcal{B}$ for anchor points, there are two strategies for selecting the anchor points in space:

- Data dependent approach. We randomly sample anchor points following the query history distribution, which boosts the accuracy of estimation in frequently queried locations.
- Data independent approach. We partition the space into $\mathcal{B}$ equal cells, and select an anchor in the centre of each cell. Denote $d_{\max}$ as the maximal distance of an anchor point from any point in its cell. We can then bound the estimation accuracy for the worst case, *i.e.,* $\exp(-\alpha d_{\max})$.

In this paper, we employ the second strategy for the general case. Given $\mathcal{B}$ anchor points, we utilize an in-memory Quadtree to index the points, in order to quickly find the anchor point that is nearest to the query location. Each anchor point $a_i \in \mathcal{L}$ corresponds to a list $L(a_i)$ which stores influence value for each node. The list is stored on disk when memory is limited. Since for each query we choose only one anchor point, we load the corresponding list when the nearest anchor point has been decided. Nodes in each list are sorted based on their influence. Because the upper (lower) bound approximation ratio for each node is the same in a list, their order of upper (lower) bound influence is the same as that in the list, *i.e.,* for $u, v \in V$ and anchor point $a_i \in \mathcal{L}$, we have $I_q^U(\{u\}) \geq I_q^U(\{v\})$ and $I_q^L(\{u\}) \geq I_q^L(\{v\})$ if $I_{a_i}(\{u\}) \geq I_{a_i}(\{v\})$.

Discussion. Anchor point based estimation provides a quick estimation of the upper and lower bound of each node's influence for any given query location with approximation ratio guarantee, and we can access nodes based on their estimated upper bound order without further processing. Nevertheless, there are three main limitations inside this approach.

- The first limitation is that the index space is large. The index space is $O(\mathcal{B}n)$. It needs to sample enough anchor points to guarantee a good estimation of the upper and lower bound for each node's influence.
- The second limitation is that when the upper and lower bound of a node u's influence are estimated, the spatial distribution of nodes that are influenced by u is ignored. This ignorance may result in the opportunity to achieve tighter bounds being missed. For instance, as shown in Figure 3, node v_4 can only influence node v_5. Given a query q, vp_1 is selected as the anchor point which is the nearest anchor point to q. Clearly, we have $I_q(\{v_4\}) > I_{vp_1}(\{v_4\})$, since all the nodes influenced by v_4 are closer to q than vp_1. If we can learn this information in advance, $I_q^L(\{v_4\})$ can be easily raised from $I_{vp_1}(\{v_4\})\exp(-\alpha d(vp_1, q))$ to $I_{vp_1}(\{v_4\})$ at least.
- The third limitation is that after choosing the anchor point, the tightness of the bounds is identical for each node, *i.e.,* each node has the same resource for bound estimation. Usually, for an insignificant node (with smaller influence), a slightly loose bounds will be powerful enough to prune it from evaluation. For example, a node u's influence is 0.001. Given a selected anchor point a_i, the estimated influence upper bound of u is 1, which means an approximation ration of 1000. Suppose there are many nodes having influence larger than 1. This very loose upper bound will be still good enough to prune u from further evaluation. However, for significant nodes, whose influence may be close to one another, therefore tighter bounds are needed to distinguish them.

Influence Region based Estimation. To alleviate the limitations in anchor point based approach, we introduce the influence region based estimation in this section, which allocates different resources based on the influence of users.

Influence Region. The motivation of influence region based approach is that instead of storing a set of anchor points for each user u, we store the influence of the nodes to each part of the space. Formally, for $u \in V$, the influence region $R(u) = \{r_1^u, r_2^u, \dots\}$ consists of a set of disjoint grids, which covers all the area that u can influence. For each $r_i^u \in R(u)$, it also records the influence of u on the nodes located in r_i^u, which is calculated as follows:

$$I(\{u\}, r_i^u) = \sum_{v \in V_{r_i^u}} I(\{u\}, v)$$

where $V_{r_i^u}$ denotes all the nodes in V that fall into r_i^u. Since each pair of cells in $R(u)$ is disjoint, we can calculate the influence of u as follows:

$$I(\{u\}) = \sum_{r_i^u \in R(u)} I(\{u\}, r_i^u)$$

Derive Bounds. Given a query location q and a grid $r_i^u \in R(u)$, $d_{\max}(q, r_i^u)$ and $d_{\min}(q, r_i^u)$ denote the maximal and minimal distance of q to in r_i^u. $d_{\max}(q, r_i^u)$ and $d_{\min}(q, r_i^u)$ can be computed in constant time. Then we have:

$$\begin{aligned} I_q(\{u\}) &= \sum_{r_i^u \in R(u)} \sum_{v \in V_{r_i^u}} I_q(\{u\}, v) \\ &= \sum_{r_i^u \in R(u)} \sum_{v \in V_{r_i^u}} I(\{u\}, v) c \exp(-\alpha d(v, q)) \\ &< \sum_{r_i^u \in R(u)} \sum_{v \in V_{r_i^u}} I(\{u\}, v) c \exp(-\alpha d_{\min}(q, r_i^u)) \end{aligned}$$

In consequence, based on the influence region, we can derive the upper bound of u's influence as follows:

$$I_q^U(\{u\}) = \sum_{r_i^u \in R(u)} I(\{u\}, r_i^u) c \exp(-\alpha d_{\min}(q, r_i^u)) \quad (6)$$

Following similar methods, we can obtain the lower bound of u's influence as below.

$$I_q^L(\{u\}) = \sum_{r_i^u \in R(u)} I(\{u\}, r_i^u) c \exp(-\alpha d_{\max}(q, r_i^u)) \quad (7)$$

Given a query q, for each node $u \in V$ we can obtain the upper and lower bound of u's influence by scanning $R(u)$ once.

Influence Region Computation and Maintenance. According to Equations (6) and (7), the tightness of the bounds is decided by $d_{\max}(q, r_i^u)$ and $d_{\min}(q, r_i^u)$. If we treat each node influenced by u as a grid itself in $R(u)$, *i.e.,* $d_{\max}(q, r_i^u) = d_{\min}(q, r_i^u)$, then we have

$$I_q^L(\{u\}) = I_q^U(\{u\}) = I_q(\{u\})$$

which means we have stored the exact influence of u. However, improving the estimation accuracy requires more space and takes longer to process. To compute the influence region, we partition the nodes influenced by $u \in V$ using the techniques for building a Quadtree. We recursively partition the nodes influenced by u in space until each cell contains at most b nodes or the tree reaches a certain hight. For each leaf node, we shrink the cell to the minimum bound box that contains the nodes in the leaf. Lastly, $R(u)$ corresponds to the shrunken nonempty leaf nodes. Thus, the size of $R(u)$ is proportional to the area influenced by u.

Discussion. Unlike anchor point based estimation, influence region based estimation can improve estimation accuracy for a significant node u by using larger $|R(u)|$. There are two main limitations to this method:

- It is time-consuming in the pre-processing phase. For each node u, we need to scan its $R(u)$ to derive the upper and lower bound. For the sake of accessing the nodes based on their influence upper bound, we need to further sort V, which requires $O(\sum_{v \in V} |R(u)| + |V| \log |V|)$ time to complete the estimation.
- Space consumption of index. The pre-computed $R(u)$ is resident in memory, and costs $O(\sum_{v \in V} |R(u)|)$ space in total, which consumes much more space than anchor point based approach.

Fused Approach. Anchor point based method can achieve a fast estimation speed, while the influence region based

method can provide a tighter bound for significant nodes. This motivates us to take advantage of both approaches by combining their positive aspects.

Derive Bounds. We maintain $\mathcal{B}'$ anchor points.

- *For less influential nodes* V_s, we use the anchor points based approach to achieve a loose upper and lower bound, and we can obtain the order of the nodes based on their upper bound efficiently.
- *For users with large influence* V_l, we store the influence region index for each of them, and we refine their influence bounds on the fly, *i.e.,* $I_q^U(\{u\}) = \min\{I_q^{Ua}(\{u\}), I_q^{Ur}(\{u\})\}$, where $I_q^{Ua}(\{u\})$ and $I_q^{Ur}(\{u\})$ are the upper bound of u derived based on anchor point based approach and influence region based approach respectively. Similarly, we refine the lower bound for $u \in V_l$ with $\max\{I_q^{La}(\{u\}), I_q^{Lr}(\{u\})\}$. After refinement, we sort nodes in V_l based on their influence upper bound.

V_s and V_l Selection. To select nodes for V_s and V_l, we utilize a threshold τ. We sort nodes $v \in V$ by their influence $I(\{v\})$; V_l consists of the *top*-τ nodes with the largest influence, and V_s consists of the rest of the nodes, *i.e.,* $V_s = V \setminus V_l$. Compared with $|V|$, τ is much smaller. Because the influence distribution of users in a social network fulfils the long tail distribution, we may expect these insignificant users take the majority of them and have little chance to get involved into the selected seed set. For example, in the experiments 300 is selected as the default value for τ.

Example 3: As shown in Figure 3, suppose we set $\tau = 1$, then v_3 is selected as the one stoing influence region index, since $I(\{v_3\})$ is the largest of the 5 nodes. We partition the nodes influenced by v_3 into three parts, and then we have $R(u) = \{\{v_1\}, \{v_2\}, \{v_4, v_5\}\}$. We can calculate the exact influence for the first two partitions online, and for the third partition, we have $d_{\min}(q, r_3^{v_3}) = d(q, v_4)$ and $d_{\max}(q, r_3^{v_3}) = d(q, v_5)$.

The space and time complexity analysis for the fused approach can be easily derived based on the previous two approaches. Using the fused approach, we can both decrease the index cost and improve the tightness of bounds for nodes with large influence.

B. Rule 2: Marginal Influence Upper Bound Estimation

For $u \in V$, estimating the upper bound of marginal influence is harder than estimating the upper bound of influence, since u may have correlation with the selected seed set S. In this section, we investigate the techniques to derive the upper bound of node u's marginal influence, used in Rule 2.

Derive Upper Bound. Given nodes $u, v \in G$, let $I(u|S, v)$ and $I_q^U(u|S, v)$ denote the marginal influence and upper bound of distance aware marginal influence from u to v. For any node $w \in MIOA(u)$, if $I(S, w) = 0$ which means there is no correlation between S and u for node w, we have $I(u|S, w) = I(\{u\}, w)$, otherwise we have $I(u|S, w) < I(\{u\}, w) - I(\{u\}, w) \times I(S, w)$. Thus we can derive $I_q^U(u|S, v)$ as shown in Equation (8).

$$I_q^U(u|S, v) = \begin{cases} I_q(\{u\}, v) & \text{, if } I(S, v) = 0 \\ I_q(\{u\}, v)(1 - I(S, v)) & \text{, otherwise} \end{cases} \tag{8}$$

Therefore, we can compute the upper bound of u's distance aware marginal influence $I_q^U(u|S)$ as follows.

$$I_q^U(u|S) = \sum_{v \in MIOA(u)} I_q^U(u|S, v) \tag{9}$$

Improvement. Using Equation (9), we can estimate the upper bound of u's marginal influence by scanning $MIOA(u)$ once. However, this procedure can be time-consuming when the size of $MIOA(u)$ is large. To alleviate the problem, we develop a partition based approach.

Compared with the initial influence, the activated probability of u is not from 0 to 1, but from $I(S, u)$ to 1, so the marginal influence of u becomes smaller than its initial influence. Without the MIA model heuristic, if S can influence u, S can also influence the nodes that u can influence. In consequence, the marginal influence of u to v can be bounded by Equation (10).

$$I_q(u|S, v) \leq (1 - I(S, u))I_q(\{u\}, v) \tag{10}$$

Note that this upper bound is different from the bound in (8). Equation (10) is based on the supposition that S will not influence the nodes in the paths from u to v. The upper bound of u's marginal influence can therefore be calculated as follows.

$$\begin{aligned} I_q^U(u|S) &= \sum_{v \in V} (1 - I(S, u))I_q(\{u\}, v) \\ &= (1 - I(S, u))(I_q(\{u\})) \end{aligned} \tag{11}$$

If we do not know $I_q(\{u\})$ when estimating $I_q^U(u|S)$, we can replace it with the upper bound developed in Section III-A.

Based on Equation (11), the estimation cost is only $O(1)$. However, this equation does not work under the MIA model. In the MIA model, S cannot influence the node $v \in MIOA(u)$, even if $I(S, u) > 0$. We have $I(S, v) > 0$, only if there is a node $s_i \in S$ that has a maximal influence path from s_i to v. For example, as shown in Figure 2, suppose $S = \{v_2\}$. Under the MIA model, we have $I(S, v_3) > 0$ and $I(\{v_3\}, v_7) > 0$, but $I(S, v_7) = 0$, since $\mathcal{P}(\langle v_2, v_3, v_7 \rangle) < 0.25$. The same holds for node v_4, so $I_q^U(v_3|S)$ still equals $I_q(\{v_3\})$. Based on the analysis, Equation (11) may return a value that is smaller than u's marginal influence, therefore we need to make some modifications to derive the upper bound.

Let $I_{\min}(S, u)$ equal $\min\{I(S, v)|v \in MIOA(u)\}$. We can then replace $I(S, u)$ with $I_{\min}(S, u)$ in Equation (11). It is easy to verify that the new derived upper bound is held under the MIA model. However, $I_{\min}(S, u)$ could be very small or equal 0, if there are nodes in $MIOA(u)$ that cannot be influenced by S. Moreover, we need to calculate $I_{\min}(S, u)$ whose cost can equal the cost of using Equation (9). To overcome the problem, we partition the nodes in $MIOA(u)$.

Definition 3 ($MIOA(u)$ Partition): Given a node $u \in V$ and a parameter $\Delta \in (0, 1)$, $MIOA(u)$ partition is defined as $PT(u) = \{pt_1^u, pt_2^u, \ldots, pt_{\lceil \log_\Delta \theta \rceil}^u\}$, which fulfils the following conditions: *1)* $pt_i^u \subseteq MIOA(u)$ and $\bigcup pt_i^u = MIOA(u)$; *2)* $pt_i^u \cap pt_j^u = \emptyset$, for $i \neq j$; *3)* $I_{\min}^i(u) \in [\Delta^{i-1}, \Delta^i)$, where $I_{\min}^i(u) = \min\{I(\{u\}, v)\}$ for $v \in pt_i^u$.

$I_{\min}^i(u)$ is fixed for each $MIOA(u)$ partition. If pt_i^u is empty, we simply remove it from the partition. For a seed

$s_i \in S$, s_i can influence all nodes in pt_i^u, if $I(\{s_i\}, u) \geq \frac{\theta}{I^i_{\min}(u)}$. Based on these criteria, we partition S into $\lceil \log_\Delta \theta \rceil$ parts for a node $u \in V$.

Definition 4 (Seed Set Partition): Given a seed set S, the partition of S corresponding to u, denoted as $PT(S, u)$, equals $\{pt_1(S, u), pt_2(S, u), \ldots, pt_{\lceil \log_\Delta \theta \rceil}(S, u)\}$, where $pt_i(S, u) = \{s_i | I(\{s_i\}, u) \geq \frac{\theta}{I^i_{\min}(u)}$ and $s_i \in S\}$.

For u, we dynamically maintain $I(pt_i(S, u), u)$ for each partition in $PT(S, u)$ when S expands in each iteration. The update can be realized by traversing the path from each new added seed to u through $MIOA(u)$. Then we can estimated the upper bound of u's marginal influence as shown in Equation (12).

$$\begin{aligned} I_q^U(u|S) &= \sum_{i=1}^{\lceil \log_\Delta \theta \rceil} \sum_{v \in pt_i^u} (1 - I(pt_i(S, u), u)) I_q(\{u\}, v) \\ &= \sum_{i=1}^{\lceil \log_\Delta \theta \rceil} (1 - I(pt_i(S, u), u)) I_q(\{u\}, pt_i^u) \end{aligned} \quad (12)$$

where $I_q(\{u\}, pt_i^u)$ denotes the total influence of u to the nodes in pt_i^u. When we do not have the exact value of $I_q(\{u\}, pt_i^u)$, we use the upper bound developed in Section III-A to replace it, which can be achieved by storing more information for each partition of $MIOA(u)$.

Index Maintenance and Discussion. In this section, we consider two approaches to obtain the upper bound of a node's marginal influence under the MIA model.

- For the approach using Equation (9), the upper bound of marginal influence is computed in $O(|MIOA(u)|)$ time, and no more extra index is needed. When $|MIOA(u)|$ is small, which means u is insignificant, we can directly use this method to estimate the marginal influence upper bound.
- For the approach based on Equation (12), the upper bound is computed in $O(\lceil \log_\Delta \theta \rceil)$ time. Note that the smallest θ we use in the experiment is 0.001, if $\Delta \leq 0.5$, we have $\lceil \log_\Delta \theta \rceil \leq 10$. However, we need to maintain $\lceil \log_\Delta \theta \rceil$ times information for the upper estimation of nodes influence, since we need to estimate the upper bound of $I_q(\{u\}, pt_i^u)$ when the exact value is unknown. To reduce the space cost, we only use this approach for the *top-*τ nodes, where *top-*τ nodes are the same as given in Section III-A.

C. Rule 3: Approximate Result Estimation

In this section, we first clarify the motivation and details about Rule 3. Then we introduce a view based approach to obtain an approximate result used in Rule 3.

Motivation and Definition of Rule 3. By applying Rules 1 and 2, we can prune many nodes from exactly evaluating their influence or marginal influence, and the final seed set obtained is still identical to that achieved using the traditional greedy framework, *i.e.,* Algorithm 1. Sometimes we are allowed to "relax" the requirement of the returned result. In this case, we can achieve potential speedup or early termination in each iteration, but the lower bound of the result's quality, *i.e.,* $1 - \frac{1}{e}$ approximation ratio, is retained. Bellow is an example to clarify the motivation of Rule 3.

Example 4: Given an influence maximization problem for $k = 2$, we initially obtain a near optimal result $\{v_1, v_2\}$, in which approximation ratio is bounded by $0.8(1 - \frac{1}{e})$. $I(\{v_1\}) = 5$ and $I(\{v_1, v_2\}) = 8$. If we can find a seed set S with influence larger than $8/0.8 = 10$, its approximation ratio must be larger than $1 - \frac{1}{e}$. Suppose for the first seed, we find the node with influence equalling $7 > 5/0.8 = 6.25$. For the second seed, we only need to find the one that has a marginal influence larger than $10 - 7 = 3$, and the approximation ratio of the final seed set found is guaranteed.

Formally, given a DAIM problem, suppose we have an ordered seed set S^v of size k which has $\beta(1 - \frac{1}{e})$ approximate ratio for any prefix of the seed set, *i.e.,*

$$I_q(S_i^v) \geq \beta(1 - \frac{1}{e}) I_q(S_i^*) \text{ for } i = 1, 2, \ldots, k$$

where S_i^v is the prefix of S^v with i nodes and S_i^* is the optimal seed set with i nodes. We can define Rule 3 with the following two rules:

Rule 3.1: Early termination when the quality lower bound is met, that is if we have $I_q(u|S_{i-1}) \geq I_q(S_i^v)/\beta - I_q(S_{i-1})$, we can terminate this iteration and select u as the next seed, where S_i is the seed set selected in the first i iterations.

Rule 3.2: The selected seed set should be better than S^v, that is if $I_q(u|S_{i-1}) \leq I_q(S_i^v \setminus S_{i-1}^v | S_{i-1})$ and $S_i^v \setminus S_{i-1}^v$ is not included in S_{i-1}, we can prune u from evaluation in this iteration.

Note that when Rule 3.1 is used to find a seed set, we need to check if $I_q(S^v)/\beta \leq I_q(S)$. Otherwise, the returned seed set S may affect the approximate ratio guarantee. When success for Rule 3.1, the influence of returned seed set may be smaller than the seed set which follows the exact greedy framework.

Derive Approximate Results. To obtain the seed set needed for Rule 3, we extend the anchor point based approach. Suppose there is a $k_{\max}$, which is the maximal number of seeds allowed to be selected by users, *e.g.,* users are only allowed to choose 100 nodes at most for product promotion.

Off-line Computation: We sample a set $\mathcal{V}$ of locations, called view points, which applies the same sampling strategy as given in Section III-A. For each location $vp_i \in \mathcal{V}$, we compute the seed set S^{vi} by taking vp_i as the query location with $|S^{vi}| = k_{\max}$. Since nodes in S^{vi} are selected one by one using a greedy strategy, any prefix of the seed set with size k' is also a solution when the seed set size is k'. The locations in $\mathcal{V}$ are maintained with a in-memory Quadtree, and the corresponding seed set and influence are stored on disk when memory is limited.

Online Processing: Given a query location q, we retrieve the nearest view point vp using the Quadtree and load the seed set S^{vp}. We calculate $I_q(S^{vp})$ and use S^{vp} as the seed set in Rule 3. Even though this method is simple, we can achieve a bounded approximation ratio, as shown in Lemma 3.

Lemma 3: $I_{vp}(S^{vp})$ achieves $\exp(-\alpha d(vp, q))(1 - \frac{1}{e})$ approximation ratio.

The lemma is correct based on Equation (5). To use Rule 3.1, we simply need to replace $I_q(S_i^u)$ with $I_{vp}(S_i^{vp})$.

Discussion. The space cost of view point based approach is $O(k_{\max}|\mathcal{V}|)$, where $|\mathcal{V}|$ is the number of view points retained. Compared with anchor point based estimation, the information stored on each point is much smaller, which gives us the opportunity to sample more view points. The pruning power of Rule 3 becomes more obvious when $|\mathcal{V}|$ is large. The intuition is that with high probability two close stores may choose the same group of people for doing local promotion.

Sometimes, we may not know $k_{\max}$ in advance. In this situation, we choose a reasonable $k_{\max}$, *e.g.,* 100. When the query $k > k_{\max}$, we can apply Rule 3 for the first $k_{\max}$ seed selection, and for the remaining $k - k_{\max}$ seeds, we apply the greedy framework to find the qualified result.

IV. Priority based Algorithm

In this section, we introduce a priority based algorithm to efficiently select k nodes for a given DAIM problem, which integrates the pruning rules proposed in Section III.

Algorithm. The basic idea is that we look up the nodes in V according to node significance order, *i.e.,* the nodes' influence or marginal influence. Based on influence bounds derived, we can consequently prune many insignificant nodes from exactly computing their influence or marginal influence in the current iteration. Algorithm 2 describes the details of the priority based algorithm, which takes a geo-social network G, a query location q and an integer k as input, and outputs a seed set of k nodes. In Algorithm 2, we utilize all the pruning rules introduced in Section III, and for Rule 1, we adopt fused approach. To switch off any pruning rule, we need to comment the corresponding pre-computed index and derived bounds.

Off-line Processing. Let V_τ denote the set of top-τ influential nodes defined in Section III-A. We pre-compute $MIIA(u)$ and $MIOA(u)$ for each node $u \in V$ (Line 1), anchor point set $\mathcal{L}$ and influence region index $R(u)$ for $u \in V_\tau$ (Line 2) used in Rule 1, $MIOA(u)$ partition for $u \in V_\tau$ (Line 3) used in Rule 2, and view point set $\mathcal{V}$ (Line 4) used in Rule 3.

For online query processing, we divide it into three phases:

Initialization. H is a priority queue which maintains the set of potential nodes. In Line 7, we derive the upper and lower bound for each node in V. For nodes in V_τ, we use the influence region index to estimate the bounds, and for the other nodes, we use the anchor point based approach. Note that when a node's exact influence is calculated, its upper bound is equals to its lower bound. We sort V based on the upper bound of node's influence decreasingly in Line 8 and insert them into Q. In Line 9 and 10, we obtain the view point vp for q and calculate $I_q(S_i^{vp})$ for $i = 1, 2, \ldots, k$. To calculate the exact influence or marginal influence of u, we follow the method in [5]. Next in Line 11, we remove S^{vp} from Q and insert them into H with the node influence as the key.

First Seed Selection. $H^L_{\max}$ denotes the largest lower bound of the node influence in H. As we have used Rule 3, $H^L_{\max}$ equals $\max\{I_q(\{u\})|u \in S^{vp}\}$ initially, otherwise it equals 0. We keep adding nodes from Q to H and updating $H^L_{\max}$, until it is larger than $Q.top$. From Lines 13 to 21, we keep evaluating the nodes in H. If $I_q(\{u\})$ is calculated, u is the node with the largest influence in H and is added to S, otherwise, we calculate $I_q(\{u\})$ in Line 18. If u fulfils the Rule 3.1 early termination criteria in Line 19, we add u to S and terminate this iteration, or u is re-inserted into H in Line 21. The UpdateState() function in the algorithm updates $I_q(S, v)$ for $v \in MIOA(u)$, and the $MIOA(v)$ partition information if $v \in V_\tau$, after selecting a new seed u.

Subsequent Seed Selection. For the rest $k - 1$ seeds selection from Line 22 to 36, it follows the similar framework as selecting the first seed. The difference is that: *1)* u is added to S, if $I_q(u|S)$ is calculated and is larger than $Q.top$ in Line 24 and 25 to guarantee u is the node with marginal influence. *2)* We can utilize the upper bound of marginal influence in Line 36 to further prune nodes based on Rule 2. Recall that when Rule 3.1 is used to find a seed set, we need to check if

Algorithm 2: Priority based Algorithm

Input : G : geo-social network, k : seed set size, q : a query location.
Output : S : seed set.

// Off-line processing: build index
1 Pre-compute $MIIA(u)$ and $MIOA(u)$ for $u \in V$;
2 Pre-compute $\mathcal{L}$ and $R(u)$ for $u \in V_\tau$;
3 Pre-compute $MIOA$ partitions for $u \in V_\tau$;
4 Pre-compute view point set $\mathcal{V}$;
// Online processing: Initialization
5 Initialize seed set: $S \leftarrow \emptyset$;
6 Initialize priority queue: $H \leftarrow \emptyset$;
7 Estimate upper and lower bound of $I_q(\{u\})$ for $u \in V$;
8 Initialize queue: $Q \leftarrow$ sorted V;
9 $vp \leftarrow$ nearest point in $\mathcal{V}$ to q;
10 Calculate influence for nodes in S^{vp};
11 $Q \leftarrow Q \setminus S^{vp}$; $H \leftarrow H \cup S^{vp}$;
// Online processing: First seed selection
12 Add nodes from Q to H until $H^L_{\max} \geq Q.top$;
13 **while** $True$ **do**
14 $u \leftarrow H.pop$;
15 **if** $I_q(\{u\})$ *is calculated* **then**
16 $S \leftarrow \{u\}$; UpdateState(); Break;
17 **else**
18 Update the key of u with $I_q(\{u\})$;
19 **if** u $fulfils$ *Rule* 3.1 **then**
20 $S \leftarrow \{u\}$; UpdateState(); Break;
21 **else** Re-insert u to H;
// Online processing: subsequent seed selection
22 **while** $|S| < k$ **do**
23 $u \leftarrow H.pop$;
24 **if** $I_q(u|S)$ *is calculated* **then**
25 **if** $I_q(u|S) \geq Q.top$ **then**
26 $S \leftarrow S \cup \{u\}$; UpdateState();
27 **else**
28 Add nodes from Q to H until $I_q(u|S) \geq Q.top$;
29 Re-insert u to H;
30 **else if** $I_q(u|S)$ *is estimated* **then**
31 Update the key of u with $I_q(u|S)$;
32 **if** u $fulfils$ *Rule* 3.1 **then**
33 $S \leftarrow S \cup \{u\}$; UpdateState(); Break;
34 **else** Re-insert u to H;
35 **else**
36 Update the key of u with $I^U_q(u|S)$; Re-insert u to H;

$I_q(S^v)/\beta \leq I_q(S)$. Otherwise, we need to restart the search without Rule 3.1 to guarantee the approximation ratio.

Discussion. In the implementation of Algorithm 2, when the estimated upper and lower bounds are not tight, it may cause loading many nodes into H in Line 12 and 28. To avoid this issue, we can periodically calculate the exact influence or marginal influence for the node at the top of H when loading nodes. For example, if the procedure in Line 12 does not stop after adding each 50 nodes from Q to H, we calculate the influence of node u' at the top H, and $H^L_{\max}$ can be refined with $\max\{I_q(\{u'\}), H^L_{\max}\}$. A similar strategy is implemented for Line 28.

As stated in Lemma 4, the quality of the returned seed set in priority based algorithm is bounded.

Lemma 4: For the DAIM problem, Algorithm 2 returns a seed set of k nodes with $1 - \frac{1}{e}$ approximation ratio under the MIA model.

Proof: When Rule 3 is not applied in the algorithm, the returned seed set is identical to that returned in Algorithm 1, since Rule 1 and Rule 2 only prune the nodes that do not belong to the seed set in each iteration. When Rule 3 is applied, it can achieve possible early termination, but the result quality is bounded as stated in Section III-C, since we need to check the relation between $I_q(S^v)$ and $I_q(S)$ in the final. ■

V. Experiments

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

A. Experiment Setup

Algorithms. We compare our algorithms with both the heuristic method and the greedy method. For the heuristic method, we extend the PMIA approach [5] to support DAIM problem to demonstrate the efficiency, *i.e.,* response time, of our approach. For the greedy algorithm, we extend the CELF++ approach [10] to evaluate the effectiveness, *i.e.,* distance-aware influence spread, of our approach. The source codes of both approaches are obtained from the authors. To summarize, the algorithms evaluated in this paper are listed as follows.

- PMIA: PMIA approach and we pre-compute $MIIA(u)$ and $MIOA(u)$ for each node $u \in V$.
- CELF++: CELF++ approach based on greedy framework. To calculate the influence spread, 10000 round simulations are run to calculate the influence spread.
- PRI: Priority based algorithm with Rule 1.
- PRII: Priority based algorithm with Rule 1 and 2.
- PRIII: Priority based algorithm with Rule 1, 2 and 3.

Dataset. To evaluate the algorithms, we use three real-world geo-social networks in this paper where users can share their check-in. *1)* Brightkite data[1], **BK** for short, consists of 58K nodes and 428K edges. *2)* Gowalla data[2], **GW** for short, consists of 197K nodes and 1.9 million edges, and serves as the default dataset. *3)* Twitter data [13], **TW** for short, consists of 554K nodes with 4.29 million edges.

Propagation Probability. We use the following two methods [5] to generate the propagation probability along the edge.

- WC model: Weighted cascade model, where the probability of edge $\langle u, v\rangle$ is set as $\frac{1}{N_v}$, where N_v is the number of incoming neighbours of v.
- TC model: Trivalency model, where each edge's probability is randomly selected from $\{0.1, 0.01, 0.001\}$, denoting the high, medium and low influence respectively.

Workload and Parameters. To evaluate the effectiveness and efficiency of proposed methods, the response time and distance aware influence spread are reported. The query locations are randomly selected from the space. The seed set size k varies from 10 to 50 with 10 as the default value. The anchor point set size $|\mathcal{L}|$ varies from 100 to 300 with 200 as the default. The view point set size $|\mathcal{V}|$ varies from 500 to 1500 with 1000 as the default. For the region based estimation, we set τ equal to 300. For the parameter in function f, c is set to 10 and α is set to 0.02. For the maximal influence path, θ is set to 0.001.

To obtain the influence spread of the heuristic algorithms, we run 10000 round simulations for each seed set and take the average of the influence spread, which matches the setting of the CELF++ approach. 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.

B. Effectiveness Evaluation

To evaluate the effectiveness of the proposed algorithms, we compare them with CELF++ and PMIA based approach on three datasets and two propagation probability models. The results are shown in Figure 4 by varying k. Since CELF++ runs too long on the last two datasets, we only report its results on the BK dataset.

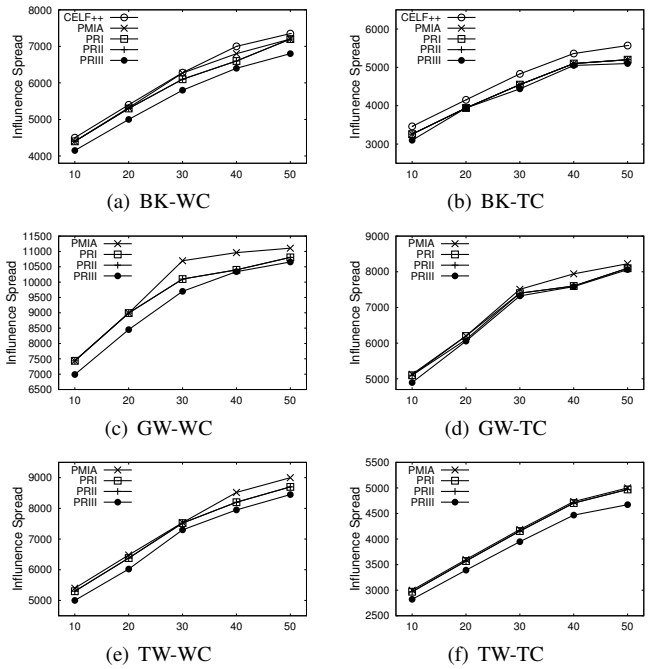
Summary of Results. *1)* The heuristic approaches achieve a slightly smaller influence spread than CELF++. *2)* The PRI and PRII approaches proposed in this paper obtain almost the same influence spread as the PMIA based approaches. *3)* The PRIII approach is only 6% smaller than PRI and PRII in influence spread for the worst case.

Figures 4(a), 4(c) and 4(e) show the results conducted on the three datasets based on WC model, and Figures 4(b), 4(d) and 4(f) show the results conducted using the TC model. As we can see, the influence spread on the TC model is smaller than that on the WC model, due to edge degree distribution. PRI and PRII have the same influence spread, since Rule 1 and Rule 2 only prune the insignificant nodes and will not affect the final seed set returned. For PRIII, the influence spread of the returned seed set is smaller than PRI and PRII, since PRIII can achieve possible early termination in each iteration and does not return the seed with maximum marginal influence when Rule 3.1 is successful, but the influence spread of the PRIII returned seed set is at most 6% smaller than that of PRI.

C. Efficiency Evaluation

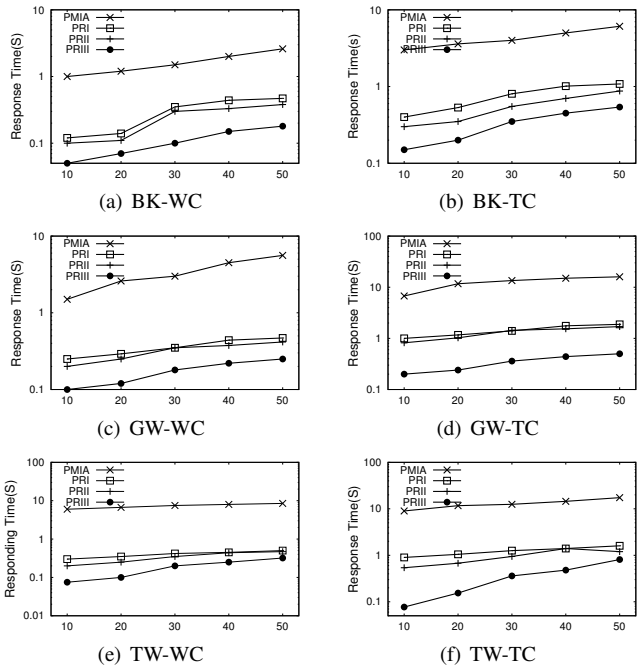
To evaluate the efficiency of the proposed algorithms and pruning rules, we report the response time by comparing with

[1]https://snap.stanford.edu/data/loc-brightkite.html
[2]https://snap.stanford.edu/data/loc-gowalla.html



(a) BK-WC (b) BK-TC (c) GW-WC (d) GW-TC (e) TW-WC (f) TW-TC

Fig. 4. Effectiveness Evaluation



(a) BK-WC (b) BK-TC (c) GW-WC (d) GW-TC (e) TW-WC (f) TW-TC

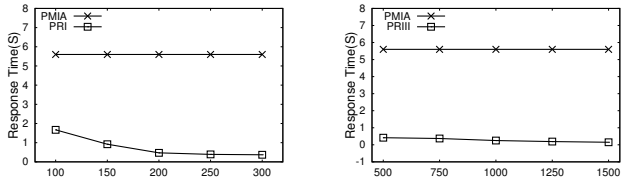
Fig. 5. Efficiency Evaluation

the PMIA based approach. Note that CELF++ is too slow, so we do not use it for comparison. Figure 5 reports the efficiency of the algorithms by varying k on three datasets, and Figure 6 shows the results when the different parameters are varied on GW dataset.

Summary of Results. *1)* PRI, PRII and PRIII greatly outperform the PMIA based approach. *2)* PRII is faster than PRI and PRIII is faster than PII. *3)* By increasing the number of anchor points and view points, we can increase the speed of the algorithms.

As shown in Figure 5, the algorithms proposed in this paper always outperform the PMIA based approach, especially PRIII which is able to achieve speedup of two order magnitudes in some cases. For the PMIA based approach, firstly, 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 obtain the influence for each node, since the weight of nodes is not known in advance. Secondly, PMIA needs to calculate the marginal influence for many nodes in each iteration, which is a big cost for online query. For the algorithms introduced in this paper, we first leverage the index for Rule 1 to obtain the lower and upper bound for each node and the initial node order, so that we can avoid touching many insignificant nodes. Secondly, based on the index for Rule 2, we reduce the number of nodes that are needed to calculate marginal influence. Thirdly, the index for Rule 3 offers the opportunity to realize early termination in each round. The results of PRI, PRII and PRIII confirm the contribution of each pruning rule. In the meantime, we can make a trade-off between effectiveness and efficiency when selecting algorithms in real application: if the response time is more important, we can use PRIII, otherwise we can adopt PRI or PRII.

In Figure 6, we conduct experiments to demonstrate the impact of $|\mathcal{L}|$ and $|\mathcal{V}|$ on GW using the WC model. As shown in Figure 6(a), the response time of PRI decreases when $|\mathcal{L}|$ is increased. This is because more anchor points means tighter bounds and the pruning power will also be improved. However, the speedup is not as significant when $|\mathcal{L}|$ is large, because the bounds estimated via anchor points are tight enough and the marginal influence calculation dominates the cost in each iteration. Similar trends can be observed in Figure 6(b) for PRIII when view point size is varied. When the size is small, the approximation ratio β in Rule 3 is too loose to provide good early termination criteria. When $|\mathcal{V}|$ is approaching infinity, the response time approaches 0, since we have indexed all the query results.



(a) GW-WC: Vary $|\mathcal{L}|$ (b) GW-WC: Vary $|\mathcal{V}|$

Fig. 6. Impact of Different Parameters

VI. Related Work

Influence Maximization in Social Network. There is a large amount of literature on the influence maximization problem [9], [17], [11], [12], [6], [5], [7], [10], [2], [19], [18]. Kempe et al. [11] 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 $1 - \frac{1}{e}$ approximation ratio using the greedy framework.

To improve efficiency and maintain the same approximation ratio, Leskovec et al. [12] authors propose a lazy forward

framework and achieve 700 times speedup compared to the naive greedy algorithm in [11]. Goyal et al. [10] further ameliorate Leskovec et al.'s methods with a 50% improvement in query time. Recently, Borgs et al. [2] propose a near-linear time approach **RIS** by sampling Random Reverse Reachable Set. Based on the same framework of RIS, Tang et al. [19], [18] further improve its efficiency in term of sample complexity and apply it to a more general diffusion model. Lucier et al. [3] propose a progressive sampling approach which can be applied to parallel framework to calculate the influence spread. In [5], 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. [6] utilized a degree discount heuristic to identify influential nodes. Chen et al. [5] 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 [7] to solve the problem under linear threshold model. In [8], 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 off-line.

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. [22] attempt to measure the influence between users by considering both social relation and location information, and aim to identify influential events. Zhu et al. [23] 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. [13], who attempt to find a seed set that will maximize the influence spread in a query region. As stated in Section I, it is non-trivial to determine an appropriate query range when conducting a location aware promotion.

VII. 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. We formally define the problem and extend the MIA model for influence approximation. To reduce the number of node evaluated, we introduce three pruning rules and novel index structures to obtain the information needed in the pruning rules. The first two pruning rules utilize the upper and lower bounds of nodes' influence and the upper bound of the nodes' marginal influence to discard insignificant nodes. Rule 3 aims to achieve possible early termination by pre-computing a set of view points. In addition, we propose a priority based algorithm which integrates the three pruning rules and returns a seed set with $1 - \frac{1}{e}$ approximation ratio under the MIA model. It is easy to turn on/off any pruning rules in the algorithm. Lastly, we demonstrate the efficiency and effectiveness of proposed techniques on three real world geo-social networks. The experiments show that our algorithms achieve high performance and maintain a large influence spread.

Acknowledgment

Ying Zhang is supported by ARC DE140100679 and DP130103245. Wenjie Zhang is supported by ARC DP150103071 and DP150102728. Xuemin Lin is supported by ARC DP150102728, DP140103578 and NSFC61232006.

References

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

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

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

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

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

[6] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, pages 199–208, 2009.

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

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

[9] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, pages 57–66, 2001.

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

[11] D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146, 2003.

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

[13] G. Li, S. Chen, J. Feng, K. Tan, and W. Li. Efficient location-aware influence maximization. In *SIGMOD 2014*, pages 87–98, 2014.

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

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

[16] J. Nail. The consumer advertising backlash. In *Forrester Research and Intelliseek Market Research Report*, 2004.

[17] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

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

[19] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD*, pages 75–86, 2014.

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

[21] Y. Wu, S. Yang, and X. Yan. Ontology-based subgraph querying. In *ICDE*, pages 697–708, 2013.

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

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