Regionally Influential Users in Location-Aware Social Networks

Panagiotis Bouros  
Department of Computer Science  
Humboldt-Universität zu Berlin  
Berlin, Germany  
bourospa@informatik.hu-berlin.de

Dimitris Sacharidis  
Institute for the Mgmt. of Information Systems  
“Athena” Research Center  
Athens, Greece  
dsachar@imis.athena-innovation.gr

Nikos Bikakis  
School of Electrical and Computer Engineering  
NTU of Athens  
Athens, Greece  
bikakis@dblab.ntua.gr

ABSTRACT

The ubiquity of mobile location aware devices and the proliferation of social networks have given rise to Location-Aware Social Networks (LASN), where users form social connections and make geo-referenced posts. The goal of this paper is to identify users that can influence a large number of important other users, within a given spatial region. Returning a ranked list of regionally influential LASN users is useful in viral marketing and in other per-region analytical scenarios. We show that under a general influence propagation model, the problem is #P-hard, while it becomes solvable in polynomial time in a more restricted model. Under the more restrictive model, we then show that the problem can be translated to computing a variant of the so-called closeness centrality of users in the social network, and devise an evaluation method.

Categories and Subject Descriptors
H.2 [Database Management]: Database Applications

General Terms
Algorithms

Keywords
location-aware services, social networks, propagation model, influence maximization, graph closeness centrality

1. INTRODUCTION

The proliferation of mobile location-aware devices (e.g., smartphones, tablets, GPS devices) and the current trend for services based upon the social interactions of their users have given rise to the so-called Location-aware Social Networks (LASN). In the most predominant LASNs, such as Foursquare or Twitter (geo-tagged tweets), a user can become friend with another, forming thus a social network, and more importantly can check-in at various places, i.e., share in public (or to her friends/followers) her current location and activity, for example eating at a restaurant, attending a concert.

In LASNs it is often useful to find users that are highly influential within a specific geographical region. Consider for example the organizer of a city-wide festival looking to attract people across city districts. The organizer could utilize a LASN to determine the most influential user within each district, and then, recruit her to locally advertise the festival. Contrary to recruiting a group of globally influential users, targeting regionally influential users has increased chances to draw attendance from all districts. As another example, consider a natural disaster, where affected people often turn to LASNs as a source of prompt information as well as a means for self-organizing community-driven help and support. The government seeking to reward the most active and helpful civilians in the aftermath, would locate the most influential LASN users within the affected area. In such scenarios, the common theme is ranking LASN users according to their computed geo-social influence.

In this work, we introduce the top-$k$ Regionally Influential LASN users (k-RIL) problem. A user checks-in at a location $\ell$ when she makes a geo-tagged post from $\ell$. Thus, given a spatial region $R$, we say that a user is regional if she has checked-in at least once at a location within $R$. Moreover, the locality of a regional user $u$ models the probability that $u$ will check-in at a location within $R$, and, in a sense, generalizes for regions the concept of “mayorship” in Foursquare. Assuming an information propagation model for social networks [7], the regional influence of a user is defined as the (expected) total locality of users she influences. Under these, k-RIL returns the $k$ regional users with the highest regional influence.

The k-RIL problem is related to the problems of Influence Maximization (IM) in social networks, and Graph Closeness Centrality (GCC). In IM, given an information propagation model, e.g., the Independent Cascade (IC) described in [7], the goal is to select a group of users, termed seeds, that collectively influence the largest number of other users. The most computationally challenging (#P-hard) task in IM problems is computing the probability of a user being influenced. Note that even though the k-RIL problem definition is similar to the case of a single seed in IM, the #P-hardness still holds. Therefore, a common strategy in IM problems is to simplify the underlying model. In this spirit, the Maximum Influence Arborecence (MIA) model [9] restricts IC with respect to the following two assumptions: (i) a user may influence another only through third users which lie on the path that maximizes the aggregate propagation probability, and (ii) only such paths with aggregate propagation probability above a pre-defined threshold are
2. PROBLEM DEFINITION

Before formally introducing the k-RIL problem, we present some necessary definitions.

Location-Aware Social Network (LASN). Let $U$ denote the set of users, $L$ the set of locations, and $C$ the set of check-ins, where a check-in $(u, l)$ means that user $u$ has checked-in at location $l$; $C(u)$ denotes the set of locations user $u$ has checked-in. The social graph $G(U, E)$ contains an (undirected) edge $(u_i, u_j) \in E$ indicating that $u_i$ and $u_j$ are friends.

Propagation Model. Each edge $(u_i, u_j) \in E$ is associated with a propagation probability $p_{ij} > 0$, which quantifies the degree of influence between the two users. This value is calculated directly from the users’ check-ins, e.g., using the Jaccard similarity:

$$p_{ij} = \frac{|C(u_i) \cap C(u_j)|}{|C(u_i) \cup C(u_j)|}$$

or it can also be determined by external parameters, such as the users’ profiles and their friendship duration.

The propagation model we adopt differs from the maximum influence arborescence (MIA) model in that it does not enforce an influence threshold (i.e., we do not require the second assumption discussed in Section 1). This model, which we refer to as MIAwoT, is a restricted version of the Independent Cascade (IC) model but is significantly less restrictive than MIA.

The concept of maximum influence paths is principle in MIAwoT. Let $\pi_{st}$ denote a simple path $u_{s[1]} \cdots u_{s[m]}$ on the social graph from user $u_{s[1]} \equiv u_s$ to $u_{s[m]} \equiv u_t$, and define the path propagation probability of $\pi_{st}$ as

$$p(\pi_{st}) = \prod_{i=1}^{m-1} P_{[i][i+1]}$$

A path from $u_s$ to $u_t$ is called the maximum influence path (mip), and denoted by $\pi_{st}^*$, if it has the highest path propagation probability among all other paths from $u_s$ to $u_t$. As there can be multiple paths that maximize the path propagation probability, the maximum influence path is selected as one of them subject to the restriction that all its subpaths are also mips (as in MIA [9]).

Users in MIAwoT, as in IC, can be in two possible states, influenced and not influenced; once a node becomes influenced it remains so. Propagation on $G$ under MIAwoT proceeds as follows. Let $S_t$ represent the set of users influenced at step $t$. At step $t = 0$, the influenced users $S_0$ are also called the seeds. Then at step $t + 1$, each user $u_i$ that was influenced at step $t$, i.e., $u_i \in S_t \setminus S_{t-1}$, may influence her neighbor $u_j$ with probability $p_{ij}$, only if edge $(u_i, u_j)$ lies on some maximum influence path starting from a seed. This last clause is what differentiates MIAwoT from IC. Note that each user is given only one chance to influence her neighbors, at the step right after she became influenced. Propagation ends at the step when no new user is influenced. We denote the set of eventually influenced users as $\Phi(S_0)$; when $S_0$ is a single user $u$ we simply denote it as $\Phi(u)$.

Problem Statement. Given a spatial region $R$, we define the set of regional users $U_R \subseteq U$ as the users who have checked-in at least once at a location inside $R$, i.e., $U_R = \{u | u \in U, \exists \ell \in C(u) : \ell \in R\}$. Moreover, for an LASN user $u$ we define the ratio $\gamma_R(u)$ of $u$’s local check-ins in $R$ over the total as the locality of the user:

$$\gamma_R(u) = \frac{|C(u) \text{ inside } R|}{|C(u)|}$$

Intuitively, the locality captures the prior probability of a user checking in at some location inside a given region $R$.

Given a region $R$, the regional influence of a user $u$ is defined as the expected sum of localities of users influenced by $u$ (expectance $E$ is on the random set $\Phi(u)$ over the propagation probabilities):

$$I_R(u) = E\left(\sum_{u' \in \Phi(u)} \gamma_R(u')\right).$$

Note that non-regional users have localities equal to zero, and thus they do not contribute to the regional influence. This means that equivalently, the summation in Equation 3 could be restricted over users in $\Phi(u) \cap U_R$.

We next state the top-$k$ Regionally Influential LASN users (k-RIL) problem.
The influence of a user \( u \) on a spatial region \( R \) is defined as the maximum impact on \( R \) under the IC propagation model, and can be represented by \( \pi_u \).

**Problem \( k \)-RIL.** Given a spatial region \( R \), return a set \( U^R_k \) of \( k \) regional users that have the highest regional influence, i.e., \( |U^R_k| = k \) and \( \forall u \in U^R_k, \forall u' \in U_R \setminus U^R_k \) it holds that \( I_R(u) \geq I_R(u') \).

**Example 1.** Figure 1 presents an example LASN of 9 users and 6 locations. Consider an 1-RIL instance with a spatial region \( R \). Figure 1a depicts the locations as diamonds, draws the region \( R \), and also shows the check-ins grouped by location. The locations inside \( R \) are drawn with filled diamonds, and their corresponding check-ins are enclosed in a box. From the check-in lists, we derive that \( u_1, u_4, u_5, u_6 \) are the regional users. Without loss of generality, we simplify formulas setting \( \gamma_R = 1 \) for all regional users. Figure 1b depicts the social graph, where regional users are shown as filled circles. The bottom right part of the figure contains the propagation probabilities \( p_{ij} \) between users \( u_i \) and \( u_j \). For the sake of the example, we assume that these probabilities are given and thus do not correspond to Jaccard similarities computed from the check-ins.

3. **Methodology**

First, in Section 3.1, we present an efficient method for computing the regional influence. Next, in Section 3.2 we outline an algorithm for solving \( k \)-RIL.

3.1 **Computing the Regional Influence**

We first prove the \#P-hardness of \( k \)-RIL under the general Independent Cascade (IC) propagation model.

**Theorem 1.** The \( k \)-RIL problem under the IC propagation model is \#P-hard.

**Proof.** The problem of computing the influence spread of a single user, which we call SIS, under the IC model is known to be \#P-hard in [9]. We reduce SIS to the \( k \)-RIL problem. Given an SIS instance, we create a \( k \)-RIL instance where the social graph is identical, the region \( R \) is equal to the entire space, \( k \) is equal to \( |U| \), and the locality \( \gamma_R \) is equal to 1 for all users. Then, it is easy to determine the answer to the SIS instance by solving the \( k \)-RIL instance. Essentially, in order to rank the users, you need to compute the regional influence of all users, which in turn means that you have solved the SIS instance, as the influence spread of a user in SIS equals its regional influence in the particular \( k \)-RIL instance.

This result justifies our adoption of a model more restricted than IC, namely the MIAwoT propagation model. Under MIAwoT, it is possible to solve \( k \)-RIL in polynomial time. To reach this conclusion, we first show that the regional influence of a user can be computed exactly using a closed form deterministic formula.

**Lemma 1.** The regional influence of a user \( u \), under MIAwoT is:

\[
I_R(u) = \sum_{u \in U_R} p(\pi_u^*) \cdot \gamma_R(u)
\]

where \( \pi_u^* \) is the maximum influence path from \( u \) to \( u_i \).

**Proof.** For any user \( u \), let \( X(u) \) be an indicator random variable such that \( X(u) = 1 \) when \( u_i \in \Phi(u) \), and \( X(u) = 0 \) otherwise. Then, Equation 3 can be rewritten as:

\[
I_R(u) = \mathbb{E}
\left( \sum_{u \in \Phi(u) \setminus U'} \gamma_R(u) \right) = \mathbb{E}
\left( \sum_{u \in U_R} X(u) \cdot \gamma_R(u) \right)
\]

Under MIAwoT, a user \( u \) can be influenced by \( u_i \) only via the maximum influence path \( \pi_u^* \) from \( u_i \) to \( u \). This means that \( u_i \) is influenced with probability equal to this path’s propagation probability \( p(\pi_u^*) \). Therefore, \( \mathbb{E}(X(u_i)) = 1 - p(\pi_u^*) = 1 - (1 - p(\pi_u^*)) = p(\pi_u^*) \), and the theorem follows.

Lemma 1 shows that the regional influence of a user \( u \), can be directly computed from the path propagation probabilities of the maximum influence paths from \( u_i \) to any other regional user. Therefore, the challenge is how to efficiently compute the propagation probabilities. Towards this goal, inspired by [9], we define a set \( W \) of edge weights for the social graph such that weight

\[
w_{ij} = - \ln p_{ij}
\]

is assigned to edge \((u_i, u_j)\). Moreover, let \( d(u_i, u_j) \) denote the social distance, i.e., the sum of weights of the shortest path on \( G \) from user \( u_i \) to \( u_j \). We emphasize that the social distance of two users is not related to the spatial locations of their check-ins and is only based on their proximity on the social graph.

**Example 2.** Returning to our running example of Figure 1, we note that the top right part of the figure shows the value of the edge weights for all propagation probability values, as computed using Equation 4. The numbers along the graph edges correspond to the weights. For illustration purposes and easy distance computations the weight values are rounded to one decimal place; we remark that this rounding does not affect the correctness of the 1-RIL result in all evaluation methods.

The following lemma shows an alternative way for computing path propagation probabilities using the edge weights.

**Lemma 2.** The path propagation probability of the maximum influence path from \( u_i \) to \( u_b \) can be computed from the social distance of \( u_i \) and \( u_b \) as:

\[
p(\pi_u^*) = e^{-d(u_i, u_b)}
\]

**Proof.** Let \( \pi_u^* = u a^1 \cdot u b^1 \cdot u a^2 \cdot u b^2 \cdots u a^t \cdot u b^t \) denote the maximum influence path from \( u_i \) to \( u_b \). Since \( p(\pi_u^*) = \prod_{k=1}^{t} p_{a^k \cdot b^k} = \exp(\ln[\prod_{k=1}^{t} p_{a^k \cdot b^k} \cdot u a^k \cdot u b^k]) = \exp(- \sum_{k=1}^{t} d(u_i, u_b)) \), the lemma follows.

Combining the results of Lemmas 1 and 2, we obtain the following formula for the regional influence of a user \( u_i \):

\[
I_R(u_i) = \sum_{u \in U_R} e^{-d(u_i, u_i)} \cdot \gamma_R(u)
\]

Moreover, it is easy to show the tractability of the \( k \)-RIL problem under MIAwoT.

**Theorem 2.** The \( k \)-RIL problem under the MIAwoT propagation model is solvable in polynomial in \( |U| \) time.

**Proof.** Computing Equation 5 for each user \( u_i \in U_R \), i.e., at most \( |U| \) times, clearly solves \( k \)-RIL. Moreover, computing Equation 5 requires finding all shortest path from \( u_i \) according to the edge weights on the social graph. This task is accomplished in \( O(|E| + |U| \log |U|) \) amortized time using Dijkstra’s algorithm, where \( |E| = O(|U|^2) \) is the number of edges in the graph. Hence, there exists an algorithm that solves \( k \)-RIL in time at most cubic in \(|U|\).

### 3.2 The Algorithm

The discussion in the previous section implies the following evaluation method to solve \( k \)-RIL, called Dijkstra-based Regional Influence Computation and denoted as DRIC. The basic idea is to compute the shortest path between any pair of regional users on the social graph. Then, for each regional user, DRIC computes her regional influence and finally, sorts the regional users according to their influence and returns the \( k \) most influential.
Algorithm 1: DRIC

Input: social graph \(G(U, E)\); set of weights \(W\); set of locations \(L\); set of check-ins \(C\); spatial region \(R\); value \(k\)

Output: top-\(k\) list \(T\)

Variables: set of regional users \(U_R\), social distance matrix \(D\)

1. \(U_R \leftarrow \text{GetRegionalUsers}(U, L, C, R)\);
2. \(\text{foreach } u_i \in U_R\)
3. \(D \leftarrow \text{Dijkstra}(u_i, G, W, U_R)\);
4. \(I_R(u_i) \leftarrow \text{ComputeRegionalInfluence}(u_i, U_R, D)\);
5. \(\text{push } u_i\) to \(T\);
6. \(\text{return } T\);

(a) Distance matrix \(D\)

(b) Regional influence

Figure 2: DRIC computations

Algorithm 1 illustrates the pseudocode of the method. DRIC receives as inputs an ASN, i.e., a social graph \(G(U, E)\) with a set of weights \(W\), a set of spatial locations \(L\) and a set of check-ins \(C\), and a \(k\)-RIL query, i.e., a spatial region \(R\) and an integer \(k\). It returns the list \(T\) of the top-\(k\) most influential regional users. DRIC utilizes two data structures: (i) the set of regional users \(U_R\) and (ii), the social distance matrix \(D\) which is a \(|U_R| \times |U_R|\) symmetric matrix that stores inside every cell \(D[u_i][u_j]\) the length of the shortest path on the social graph \(G(U, E)\) between regional users \(u_i\) and \(u_j\), i.e., \(D[u_i][u_j] = d(u_i, u_j)\).

In the beginning, DRIC invokes the \(\text{GetRegionalUsers}\) function to define the set of regional users \(U_R\) (Line 1). For every user \(u_i\) of \(U_R\) the total number of her check-ins inside \(R\) is also calculated to determine her locality \(\gamma_R(u_i)\). The implementation details of \(\text{GetRegionalUsers}\) are outside the scope of this paper; any index for spatial range queries, e.g., the R-tree [5], can be employed. Then, in Lines 2–5, the algorithm examines every regional user \(u_i\) in \(U_R\) to calculate her regional influence \(I_R(u_i)\) calling functions \(\text{Dijkstra}\) and \(\text{ComputeRegionalInfluence}\), and inserts \(u_i\) into list \(T\). \(\text{Dijkstra}\) computes the shortest path from \(u_i\) to all regional users in \(U_R\) and stores its length inside the social distance matrix \(D\), while \(\text{ComputeRegionalInfluence}\) computes \(I_R(u_i)\) using Equation 5 and matrix \(D\).

Example 3. In the 1-RIL example of Figure 1b, there exist 4 regional users, \(u_1, u_4, u_5, u_6\). DRIC calls \(\text{Dijkstra}\) once for each regional user to compute entries of the social distance matrix \(D\), one row at a time. The resulting matrix is shown in Figure 2a. After each \(\text{Dijkstra}\) invocation, DRIC computes the regional influence of the examined user from Equation 5. Consider user \(u_1\) for example. Her influence is \(I_R(u_1) = e^{0.2} + e^{-1} + e^{-0.4} + e^{-1.1} = 1 + 3/8 + 2/3 + 1/3 = 2.375\). Finally, after all regional influences are computed, depicted in Figure 2b, DRIC returns \(u_5\), having the highest regional influence, as the answer to 1-RIL.

Complexity. The DRIC algorithm performs exactly \(U_R\) iterations. Each iteration invokes Dijkstra’s algorithm, which performs \(|E|\) edge relaxations and \(|U|\) dequeue operations. Assuming a Fibonacci heap, each of these operations require \(O(1)\), and \(O(\log |U|)\) amortized time. Note that an iteration also computes the regional influence, which however takes \(O(|U|)\) time and is thus dominated by Dijkstra’s running time. Therefore, the total (amortized) running time of DRIC is \(O(|U_R||E| + |U_R||U| \log |U|)\).

4. EXPERIMENTS AND CONCLUSIONS

We finally present a preliminary experimental evaluation of our methodology for identifying the top-\(k\) regionally influential users.

Table 1: Datasets characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Gozwalla</th>
<th>Brightkite</th>
<th>Foursquare1</th>
<th>Foursquare2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>197K</td>
<td>58K</td>
<td>18K</td>
<td>11K</td>
</tr>
<tr>
<td>Edges</td>
<td>950K</td>
<td>214K</td>
<td>16K</td>
<td>47K</td>
</tr>
<tr>
<td>Locations</td>
<td>1.3M</td>
<td>773K</td>
<td>43K</td>
<td>187K</td>
</tr>
<tr>
<td>Check-ins</td>
<td>6.4M</td>
<td>4.5M</td>
<td>2M</td>
<td>1.4M</td>
</tr>
</tbody>
</table>

Table 2: Response time (sec) varying query selectivity, \(k = 5\)

| \(|U_R|/|U| (\%)\) | Gozwalla | Brightkite | Foursquare1 | Foursquare2 |
|-----------------|-----------|------------|------------|------------|
| 0.1 | 1.408 | 9.6 | 0.9 | 0.2 |
| 0.2 | 262.1 | 17.9 | 2.3 | 0.4 |
| 0.3 | 432.5 | 26.8 | 3.2 | 0.6 |
| 0.5 | 590.6 | 42.1 | 5.8 | 0.9 |
| 1 | 1148.6 | 71.5 | 11.2 | 1.9 |

Our analysis involves 4 datasets from real-world LASNs. Table 1 summarizes the characteristics of these datasets. The evaluation is carried out on an 2.67Ghz Intel Xeon CPU E5640 with 32GB of RAM running Debian Linux and DRIC was written in C++. To assess the performance of DRIC algorithm, we measure its average response time over 500 \(k\)-RIL queries, varying query selectivity \(|U_R|/|U|\), i.e., the number of regional users over the total number of LASN users. Note that we choose to directly vary the selectivity of a query instead of the size of its spatial region \(R\) as the most time consuming step of the method (the Dijkstra algorithm) is related to the number of users checked-in at a location inside \(R\) and not to how large this region is.

Table 2 reports the response time of DRIC while varying query selectivity \(|U_R|/|U|\). The results verify the complexity analysis of Section 3.2. The response time increases linearly to the number of regional users \(|U_R|\). Naturally, DRIC slows down with the increase of the size of the social graph. Note that the performance of DRIC is not affected by the number of returned users, so \(k\) is set to 5.

DRIC can efficiently solve \(k\)-RIL in case of small size social networks or small number of regional users. Motivated by this, in the future we plan to devise more efficient methods that will avoid computing the influence for all regional users by examining them in descending order of their expected influence.

Acknowledgements This research was partially supported by the German Research Foundation (DFG) through the Research Training Group METRIK, grant no. GRK 1324, and by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

5. REFERENCES