Influential Nodes in a One-Wave Diffusion Model for Location-Based Social Networks[*]

Hao-Hsiang Wu[1,2] and Mi-Yen Yeh[2]

[1] Graduate Institute of Networking and Multimedia,
National Taiwan University, Taipei, Taiwan
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan
{haohsiangwu,miyen}@iis.sinica.edu.tw

Abstract. Taking the Foursquare data as an example, this paper investigates the problem of finding influential nodes in a location-based social network (LBSN). In Foursquare, people can share the location they visited and their opinions to others via the actions of checking in and writing tips. These check-ins and tips are likely to influence others on visiting the same places. To study the influence behavior in LBSNs, we first propose the attractiveness model to compute the influence probability among users. Then, we design a one-wave diffusion model, where we focus on the direct impact of the initially selected individuals on their first degree neighbors. Base on these two models, we propose algorithms to select the k influential nodes that maximize the influence spread in the complete-graph network and the network where only the links with friendship are preserved. We empirically show that the k influential nodes selected by our proposed methods have higher influence spread when compared to other methods.

1 Introduction

Due to the advances in wireless communication and positioning technologies, people can surf the Internet and share their locations through mobile devices almost anytime, anywhere. This fosters the emergence of location-based social networks (LBSNs), where people can interact with each other while sharing their location information. Example applications include Foursquare[1] and Gowalla[2]. The main difference between an LBSN and a general social network is that the former introduces a new dimension of physical location that brings social networks to reality and bridges the gap between the physical world and online social networking services [1]. In an LBSN, the act of users sharing their current locations is called *check-in*. By a check-in action at certain locations, people can

[*] The research in this paper was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC100-2221-E-001-023 and NSC101-2221-E-001-013. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agency.

[1] https://foursquare.com

[2] http://blog.gowalla.com

J. Pei et al. (Eds.): PAKDD 2013, Part II, LNAI 7819, pp. 61–72, 2013.

also associate it with other additional information such as their comments about the place, the visiting time and their companions.

Prior studies on an LBSN usually focus on the human movement behavior analysis by mining their trajectories of visited locations, such as user movement prediction [2,3] and travel recommendations [4,5]. As an LBSN is also a kind of social networks that is currently a popular medium for people to share location information, we are interested in how people influence each other on the check-in behavior, how people will be attracted by other's comments on the shared locations and who are potentially influential in an LBSN to spread the location information. The answers of these problems are important to the location-based advertising applications because they can help enlarge the visibility and adoption of the products they promote. To the best of our knowledge, none of the existing works discuss finding influential nodes in an LBSN.

In this paper, taking the Foursquare data as an example, we propose to find the influential nodes in an LBSN. Foursquare is an LBSN application that provides a platform for users to share with friends or the public about their locations, which is called *venues*, by doing the check-in action through any GPS-equipped handhold devices. In addition, users can write comments, which is called *tips*, for each venue. By viewing the tips of others, a user is possibly attracted by some of them. Moreover, each user can add their interested tips to his/her *todo* list, and mark them as *done* if they did visit the corresponding venues. This information is useful for us to inference the potential influential users of the entire network especially when we do not have other explicit information of the influence behavior due to the privacy issues of Foursquare. Now, our challenges of finding influential nodes in an LBSN become the following two: How to leverage the available information to inference the influence probability between users? How to find the influential nodes under a suitable information diffusion model?

We begin with the modeling of the influence probability among users. To be more specific, we compute the likelihood that user u_i is attracted by user u_j's tips according to the proposed *attractiveness model*, which is based on the popularity of the mutual venues they have been visited and the popularity of tips written by u_j. In addition, we design the *one-wave diffusion model*, where we focus on the direct impact of the initially selected individuals on their first degree neighbors. With the attractiveness model and the one-wave diffusion model, we further design algorithms to select k influential nodes that maximize the influence spread in the complete-graph network, where a link weighted by an influential probability is built between every pair of nodes. In addition, to scale down the search space, we also consider to find influential nodes in the friendship network, where only the link between two nodes having friendship is left, and compare the results to the nodes found in the complete-graph network. By collecting the historical tip data of Foursquare, we evaluate the effectiveness of proposed models and algorithms. We report our findings on the influential nodes found in the tip data of New York and Los Angeles. The results show that the influence spread of the k nodes found in the friendship network is very close to the spread of those found in the complete-graph network under our attractiveness model and one-wave diffusion model.

The remainder of the paper is organized as follows. We discuss the related work in Section 2. In Section 3, we introduce the attractiveness model to compute the influence probability between users. Section 4 presents the one-wave diffusion model and our proposed algorithms to find the k influential nodes in LBSNs. We report the experiment results in Section 5. Section 6 concludes the paper.

2 Related Work

The influence maximization is to find a set of k nodes that maximize the information spread in a social network under some information diffusion model. Domingos and Richardson [6] were the first to study the influence maximization problem to analyze the value of customers in business. Kempe et al. [7,8] formulated the problem as an optimization problem and proved it is NP-hard under the linear threshold model and the independent cascade model. Prior studies about finding influential nodes focused on general social networks without considering the features of location-based social networks. To the best of our knowledge, we are the first to find influential nodes in LBSNs.

The linear threshold (LT) model and the independent cascade (IC) model are two generally studied information diffusion models. Kempe et al. [8] gave a comprehensive concept of these two models. Essentially, a node can be active or inactive. An inactive node may be influenced by any of its active neighbor according to a weight between them. The diffusion process starts with a set of active nodes while all other nodes are inactive. At each step, an active node remains active and the inactive one can become active only when the the total weight of its active neighbor exceeds a pre-selected threshold between 0 and 1. On the other hand, the independent cascade model works as follows. At each step of the diffusion process, only the newly active node has a chance to influence each of its inactive neighbors with a diffusion probability. When the inactive node is influenced successfully, it becomes active in the next step. Once an active node has tried to influence its neighbors in some step, it can never influence others in the following steps.

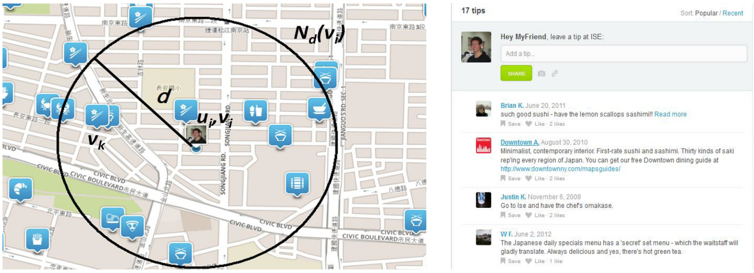
Note that both the LT and IC models consider multiple waves of influence propagation from a node to the entire network. In contrast, our proposed diffusion model only considers one wave of the influence between any initially selected node to its neighbors. In other words, the diffusion process of our proposed diffusion model involves only one step. It is because we only care the direct influence of a node to others but not the second-hand influence.

3 Modeling Attractiveness between Users

3.1 User Scenario of Foursquare

Foursquare is an location-based social networking application that provides a platform for users to share their locations, by doing the *check-in* action, with friends through any GPS-equipped handheld devices. A location is called *venue* in Foursquare. By locating the current position of a user as the center, the Foursquare application will provide the venues that fall in the neighborhood within some radius d, as shown in Fig. 1(a). All these venues are contributed by the Foursquare users and verified by the Foursquare administrators. Each venue on the map is marked by an icon showing its category. There are nine main types of categories: "Arts & Entertainment", "College & University", "Food", "Professional & Other Places", "Nightlife Locations", "Residences", "Great Outdoors", "Shops & Services" and "Travel & Transport".

In addition to check in at some venue, users can also write *tips*, namely the review comments, about the venue. Due to the interface design of Foursquare, when a user writes a tip of a certain venue, all other tips of the same venue will be listed out at the same page as shown in Fig. 1(b). Therefore, we can assume that the user will definitely be attracted by some of them. Note that these tips may come either from the friends or non-friends of the user. As the Foursquare application provides the venues within the neighborhood of radius d, the tips of these venues will also have chances to be seen by the user. In other words, any user may be attracted by the tips of venues in the neighborhood of radius d centered at his/her current location. Finally, when viewing the tips left by others, a user can add any interesting tips into his/her *todo* list. The user can further mark each todo tip to a *done* status if he/she completes the visit to the corresponding venue.



17 tips

Sort: Popular / Recent

Hey MyFriend, leave a tip at ISE:

Add a tip...

SHARE

Brian K. June 20, 2011
such good sushi - have the lemon scallops sashimi!! Read more
Save Like · 2 likes

Downtown A. August 30, 2010
Minimalist, contemporary interior. First-rate sushi and sashimi. Thirty kinds of saki rep'ing every region of Japan. You can get our free Downtown dining guide at http://www.downtownny.com/mapsguides/
Save Like · 2 likes

Justin K. November 6, 2009
Go to Ise and have the chef's omakase.
Save Like · 2 likes

W F. June 2, 2012
The Japanese daily specials menu has a 'secret' set menu - which the waitstaff will gladly translate. Always delicious and yes, there's hot green tea.
Save Like · 1 like

(a) Venues of the same category in the d-neighborhood of v_i

(b) The tip list of some venue v_k

Fig. 1. User scenario of Foursquare

3.2 The Attractiveness Model

By collecting the Foursquare data, we would like to study the influence relationship among the users and build a model to study the influential power among users in the location-based social network. In our attractiveness model, an LBSN is modeled as a graph $G = (V, E)$, where V denotes the set of nodes representing users, E is the link set representing the weighted connections between any two users. To be more specific, given two users u_i and u_j, the weight $w_{i,j}$ of the link between them is the likelihood that u_i's behavior on visiting some venues will be attracted by u_j's activities in Foursquare. In other words, $w_{i,j}$ is the influence probability of user u_j to user u_i. In the following, we show how to compute the $w_{i,j}$ value.

First, we introduce a tip and its attributes we can collect from Foursquare.

Definition 1 (A tip and its attributes). *A tip, denoted as s, has the following attributes: the user who writes this tip s.u, the category s.c, the recorded time s.t, the corresponding venue s.v, and the sum of the number of todos and the number of dones s.tdsum.* ■

As the act of adding a tip written by others into the todo list and marking it as done can be regarded as a positive feedback of a user to that tip [10], we can use this information to model how likely the later tips are attracted by the earlier ones. Due to the privacy settings of Foursquare, we cannot access the todo/done list of each user. Therefore, we cannot inference the attractiveness between users directly. However, for each tip, we can know its total number of todos and dones added and marked by different users. If a tip has a large number of todos and dones, it means that it is focused by a lot of people. Implicitly, it also shows that the user who writes this tip has some influential power.

According to the user scenario introduced in Section 3.1, a user u_i, who visits venue v_i and writes a tip for it at some time t_i, may have a chance to see other tips existing before t_i and be attracted by some of them for venues of the same category within the neighborhood of radius d centered at v_i. We denote all the venues in the d-neighborhood of v_i and having the same category of v_i as $N_d(v_i)$. Suppose g_k represents the probability that user u_i reads the tip list of venue $v_k \in N_d(v_i)$, as shown in Fig. 1(b). Among all the tips for v_k, some of them existing before t may be written by user u_j. As a result, user u_i may have a probability p_{jk} to be attracted by these tips. Then, we can compute the attractiveness of u_j to u_i, i.e., u_i is attracted by at least one tip written by u_j, for u_i to write a tip at v_i as follows.

$$P(u_i \rightsquigarrow u_j, N_d(v_i)) = 1 - \prod (g_k * (1 - p_{jk})). \tag{1}$$

Note that the above equation is under the convenient assumption that p_{jk} and g_k are independent for different v_k and different user u_j.

To compute g_k in Eq.(1), our intuition is that if a venue is hot and has a high chance to be viewed by a user, it may have a lot of tips, many of which certainly have a high number of todos and dones. As a result, we compute g_k as follows.

$$g_k|_{v_k \in N_d(v_i)} = \frac{\sum_{s \in S(v_k)} s.tdsum}{\sum_{s \in S(N_d(v_i))} s.tdsum}, \tag{2}$$

where $S(v_k)$ refers to all the tips for v_k that are written before u_i writes a tip s_i for v_i at time t_i and $S(N_d(v_i))$ refers to all the tips written before t_i for all venues in $N_d(v_i)$. Note that in case there are tips with zero sum of todos and dones, we can add one to *s.tdsum* of every tip in advance.

Next, we compute p_{jk} in Eq.(1) as follows.

$$p_{jk} = \frac{\sum_{s \in S_j(v_k)} s.tdsum}{\sum_{s \in S(v_k)} s.tdsum}, \quad (3)$$

where $S_j(v_k)$ refers to tips in $S(v_k)$ that are written by u_j. The intuition of this equation is that if a user u_j has high attractiveness, not only the number of tips that u_j writes is high but also the sum of todo and done numbers of these tips are high.

Finally, the value $w_{i,j}$, which is the influence probability of u_j to u_i, is computed as the expected value of $P(u_i \rightsquigarrow u_j, N_d(v_i))$ defined in Eq.(1).

$$w_{i,j} = \sum_{s \in S_i} P(u_i \rightsquigarrow u_j, N_d(v_i)), \quad (4)$$

where S_i are tips written by user u_i in our collected data set.

It is noted that if $w_{i,j} > w_{i,k}$, then user u_j is more attractive to user u_i compared to user u_k. In addition, when collecting the tip data of Foursquare, we can also know if two users are friends. Foursquare regularly recommends users for the tips of their friends. Also, users can actively search the tips written by their friends for venues adjacent to their interested places. Therefore, if $P(u_i \rightsquigarrow u_j, N_d(v_i)) > 0$ and u_i and u_j are friends in Foursquare, we set $P(u_i \rightsquigarrow u_j, N_d(v_i)) = 1$. This is to model that tips written by user u_i must be attracted by those written earlier by u_i's friend u_j.

4 Finding Influential Nodes

After introducing the attractiveness model, in this section, we propose *one-wave diffusion model*, which is used to model the tip information diffusion, followed by the algorithm of finding the k influential nodes that maximize the influence spread in an LBSN.

4.1 One-Wave Diffusion Model

In our attractiveness model, we show how to compute $w_{i,j}$, which is the attractiveness, or the influence power of u_j to u_i and defined in Eq.(4). Before running the diffusion process, we do normalization for these values and obtain the diffusion probability from u_j to u_i as follows.

$$q_{i,j} = \frac{w_{i,j}}{w_{max} + w_{min}}, \quad (5)$$

where w_{max} and w_{min} are the maximum and the minimum $w_{i,j}$ of the whole network.

Algorithm 1 BaseLine (G, k)

```
Input: $G = (V, E)$, number of seeds $k$
Output: A set $R$ of $k$ influential nodes
1: $R = \emptyset$
2: for $i = 1$ to $k$ do
3:     for each node $v \in V \backslash R$ do
4:         $IS_v = InfluenceSpread(v)$ /* A node will not be considered to be influenced
           by $v$ if that node is already influenced by nodes in $R$.*/
5:     $R = R \cup argmax_v\{|IS_v|, v \in V \backslash R\}$
6: return $R$
```

The diffusion process of the one-wave diffusion model works as follows. Given a start node, say v_j, it influences each of its neighbors v_i if $q_{i,j} > 0$. The total number of the influenced nodes is regarded as the influence spread of v_j. Please note that, for k seed nodes, we run the above process sequentially. Also, once a node is influenced by some seed, it cannot be further influenced by other seeds. Finally, instead of using the well-know IC or LT models in this study, our intuition of adopting the one-wave diffusion model is that we care only the influence spread of the initially selected seeds, but not that of the active nodes influenced by the seeds.

4.2 Algorithms for Influence Maximization

Given the one-wave diffusion model, in this section we show how to select k influential nodes that maximize the influential spread. Kempe et al. [8] has described a greedy algorithm to solve the k-seed selection problem, which we modify based on our one-wave diffusion model as shown in Algorithm 1.

In Algorithm 1, the *InfluenceSpread* function computes the influence spread of v according to the one-wave diffusion process and put the v with the largest influence spread into R as the solution. If there is a tie at line 5, the node with the smaller ID number wins. Because this is an exhausted search on a complete graph G (i.e., $O(|V|^2)$ links), the BaseLine approach is regarded as the benchmark but lacks efficiency. Instead, we seek if we can search influential nodes in a smaller space.

One method is called GreedyAlgorithmOnFriends(GAOF) as shown in Algorithm 2. In GAOF, we search the influential nodes on a graph where only the friendship links are retained. To be more specific, we collect the friendship between each pair of users from the Foursquare data, and remove the edge between two nodes who do not have friendship from G to generate $G_f = \{V_f, E_f\}$. First we compute the influence spread of each node, denoted as IS_v, in G_f(Lines 2-3). Finally, we extract the k nodes as the result of GAOF(Lines 4-5). If there are tie at line 5, the node with the smaller ID number wins.

Algorithm 2 GAOF (G_f, k)

```
Input: $G_f = (V_f, E_f)$, number of seeds $k$
Output: A set $R_{GAOF}$ of $k$ influential nodes
$R_{GAOF} = \emptyset$
for each node $v \in V_f$ do
    $IS_v = InfluenceSpread(v)$ /* each node $v$ has its own $IS_v$ */
for $i = 1$ to $k$ do
    $R_{GAOF} = R_{GAOF} \cup argmax_v\{|IS_v|, v \in V_f \backslash R_{GAOF}\}$
return $R_{GAOF}$
```

Algorithm 3 CGAOF (G_f, k)

```
Input: $G_f = (V_f, E_f)$, number of seeds $k$
Output: A set $R_{CGAOF}$ of $k$ influential nodes
 1: $R_{CGAOF} = \emptyset$
 2: $R = \emptyset$
 3: for each node $v \in V_f$ do
 4:     $IS_v = InfluenceSpread(v)$ /* each node $v$ has its own $IS_v$ */
 5:     compute $a_v$ of node $v$
 6: for $i = 1$ to $2k$ do
 7:     $R = R \cup argmax_v\{|IS_v|, v \in V_f \backslash R\}$
 8: for $i = 1$ to $k$ do
 9:     $R_{CGAOF} = R_{CGAOF} \cup argmax_v\{a_v, v \in argmax_v\{|S_v|, v \in R \backslash R_{CGAOF}\}\}$
10: return $R_{CGAOF}$
```

An further improvement of GAOF is called ClassifyingGreedyAlgorithmOnFriends(CGAOF) as shown in Algorithm 3. Similar to GAOF, we compute the influence spread of each node in G_f(Lines 3-4). However, in contrast to GAOF extracting k nodes by using the influence spread only, CGAOF extracts k nodes by considering three extra criteria: $|IS_v|$, $|S_v|$ and a_v, which denotes the number of influence spread of v, the number of tips written by v, and

$$a_v = \frac{\sum_{i \in V_f}\{w_{i,v} \mid \text{node } i \text{ is influenced by node } v, i \neq v\}}{|IS_v|}, \tag{6}$$

respectively. We set $a_v = 0$ when $|IS_v| = 0$. If a node v has a high a_v value, it means the attractiveness of this user to his/her friends is higher. In CGAOF, we extract the nodes by comparing $|IS_v|$ and $|S_v|$ (Lines 8-9). If there is a tie at line 7 and 9, the node with the smaller ID number wins. The intuition of this algorithm is that we think high $|S_v|$ and high a_v are also important features to compare the influence power of two nodes having only a small difference in influence spread. The node with high $|S_v|$ will have more chances to influence other nodes while the node of a_v implies this user usually writes more attractive tips.

4.3 Complexity Analysis

In this section, we analyze the complexity of each algorithm. The complexity of the one-wave diffusion process of a node v, that is $InfluenceSpread(v)$, is $O(D_v)$, where D_v denotes the degree of v in G. Thus the complexity of BaseLine is $O(k|V|D_v)$, where it extracts k influential nodes and runs $|V|$ times of the diffusion process for k nodes and selects the node with the largest influence spread. The complexity of getting R_{GAOF} from Algorithm 2 is $O(|V|D_{vf}+k|V|)$, where D_{vf} denotes the degree of some node v in G_f, and the algorithm extracts k nodes with the highest influence spread in $k|V|$ time (Algorithm 2: Line 4, 5). The complexity of getting R_{CGAOF} from Algorithm 3 is $O(|V|D_{vf} + k|V| + k^3)$, where it selects $2k$ nodes with the highest influence spread in $O(k|V|)$ and selects the highest $|S_v|$ from $2k$ nodes, then selects the highest a_v nodes from these the highest $|S_v|$ nodes, and continuously runs the selecting process k times (Algorithm 3: Line 8, 9). Because G_f only reserves the edges between two nodes that have friendship from G, $D_{vf} \ll D_v$. As a result, the time cost of GAOF and CGAOF is much smaller compared to BaseLine when $|V| \gg k$.

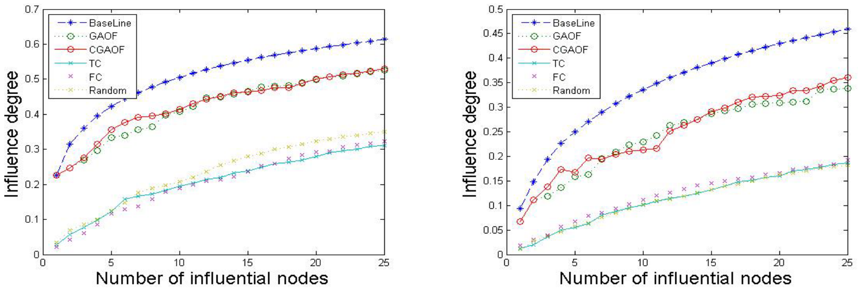
5 Performance Study

We conducted experiments on two real data sets collected from Foursquare to evaluate the effectiveness of our proposed models and algorithm. All experiments were run on a workstation with an Intel Xeon E5530 2.40 GHz CPU and 70GB RAM using C.

5.1 Settings

By crawling the Fousquare data, we obtained 47218 users who wrote tips for the venues in New York City (NYC) and 30196 users who wrote tips for the venues in Los Angeles City (LAC). There were about 410,000 tips from 2008/5 to 2011/7 in the NYC dataset and about 260,000 tips from 2009/2 to 2011/7 in the LAC dataset. The d used for $N_d(v_i)$ was set to 1 KM.

We compared the proposed three methods, BaseLine, GAOF, CGAOF, with three other methods: FriendCentrality (FC), TipsCentrality (TC), and Random on selecting the k seeds for influence maximization. We compute the influence degree, the number of influenced nodes divided by the total number of nodes in a network, of the k seeds selected by different methods. The FC and TC methods always selected k nodes having the largest number of friends and the largest number of tips, respectively, as the seeds. For the Random method, it just randomly selected k nodes as seeds. We run FC, TC, and Random several times to get their average influence degree. Please note that BaseLine, TC, FC and Random methods selected seeds from the complete network G. Only GAOF and CGAOF select k seeds from the network with only friendship links existed, i.e., G_f. In addition, for all methods, only users writing at least 50 tips were considered as the candidate of seeds. There were 754 candidates nodes in LAC and 1137 in NYC. This was to reduce the search space and speed up the initial seed selection in the two large networks.



(a) NYC

(b) LAC

Fig. 2. The influence degree of each algorithm

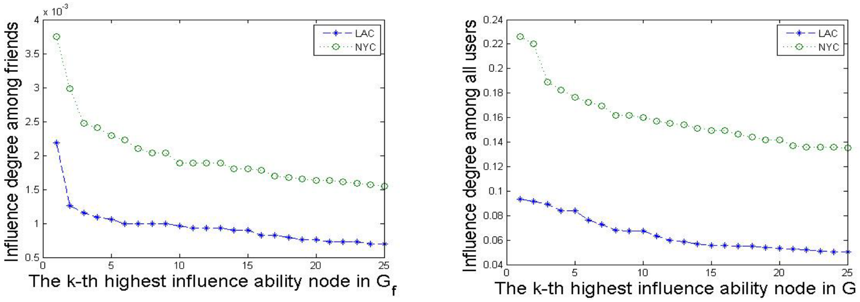
5.2 Degree of Influence Maximization

Fig. 2 shows the influence degree of the 25 seeds chosen by different methods. Note that although the five approaches selected different sets of k seeds from either G or G_f, the influence degrees here were examined for each set of k on G.

In general, the influence degree increased as the number of seeds increased on both data sets. The influence degrees of Baseline were the highest because the influential nodes were selected on the whole network G. The influence degrees of GAOF and CGAOF were lower compared to Baseline, but still of high enough values. On the NYC data set as shown in Fig. 2(a), the influence degree of the 25 nodes selected by CGAOF was high to 86.3% of that of Baseline, and that of GAOF was high to 85.5%. On the LAC data set as shown in Fig. 2(b), the influence degree of the 25 nodes selected by CGAOF was high to 78.5% of that of the BaseLine method and that of GAOF was high to 73.5%. TC, FC, and Random had poor performance on the influence degrees for both data sets showing that users who wrote the most tips or had the most friends were not necessary the most influential nodes. Instead, finding the influential nodes on G_f was a good alternative when the time cost was an primary issue. Finally, when k increased from 1 to 25, we found that CGAOF was better than GAOF in general because it considered both the number of tips written and the influential spread of a user simultaneously. Sometimes CGAOF was worse probably because G_f had some users writing many tips to attract their friends while these tips might not attract non-friends in G, where the influence degree was examined.

5.3 Independent Influence Spread among Friends and All Nodes

Here, we run the $InfluenceSpread(v)$ of each v independently on G using the Baseline approach and on G_f using GAOF. We selected the top 25 nodes of the highest influence degree values. Fig. 3(a) and 3(b) show the influence degrees of



(a) Influence among friends

(b) Influence among all users

Fig. 3. The influence ability of each node in LAC and NYC

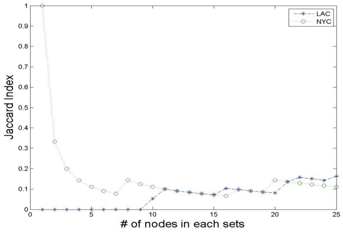


Fig. 4. Similarity between the set of the top k highest influence ability nodes in G and the set of the top k highest influence ability nodes in G_f

	LAC	NYC
# of candidate nodes	754	1137
Avg. # of tips of candidate nodes	102.2	97.9
Avg. degree per node in G	30195	47217
Avg. degree per node in G_f	28.7	48.9

Fig. 5. The statistics of G_f and G

the top 25 nodes selected from G_f and G, respectively, on both the NYC and LAC data sets. We denoted the set of 25 nodes on G_f as A and that on G as B and computed the Jaccard index $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. The higher the Jaccard index represented that the two sets were more similar to each other.

Fig. 4 shows the Jaccard indexes of A and B on both the NYC and LAC data sets. We observed there were overlaps between the two sets. Moreover, the top 1 influential node were even the same for the NYC data. We show the statistics of G_f and G in Fig. 5. It shows that although the average number of degree per node in G_f was much smaller than that of G, G_f was still useful to find the subset(about 20%-30% with k=25 in our experiments) of the top k highest influence ability nodes in G. As we mention in Section 4.3, $D_v \gg D_{vf}$ and $|V| \gg k$ in our experiments, so it was very useful to improve the efficiency. Due to our attractiveness model, if a node has friends rarely or never wrote tips, then the node could hardly or not influence its friends. Fig. 5 shows that the candidate nodes in NYC and LAC wrote the same number of tips, but NYC had a higher average number of D_{vf} compared to LAC. This means that the average number of friends of users in the LAC data was smaller compared to the NYC data such that the influence degree among friends in LAC was smaller. We also think that the influence degree may become a reason that if users in G have more friends writing tips in the same area, then it should be more efficient to use the G_f to find the influential nodes in G.

6 Conclusions

In this paper, taking the Foursquare data as an example, we studied the problem of finding influential nodes in location-based social networks. Based on the popularity of the mutual venues visited and the popularity of the tips written, we built the attractiveness model to compute the influence probability between two users. Furthermore, a one-wave diffusion model was designed to focus the direct impact of the initial seeds on their first degree neighbors. With these two models, we proposed algorithms to find the k influential nodes in LBSNs, on both a complete-graph network and a friendship network. Under our attractiveness and one-wave diffusion models, we empirically showed that the k influential nodes selected by our proposed methods in the the complete-graph and friendship networks had higher influence spread when compared to other methods.

References

1. Zheng, Y., Zhou, X.: Computing with Spatial Trajectories (2011)
2. Cho, E., Myers, S.A., Leskovec, J.: Friendship and Mobility: User Movement in Location-Based Social Network. In: Int. Conf. on KDD, pp. 1082–1090 (2011)
3. Noulas, A., Scellato, S., Mascolo, C., Pontil, M.: An Empirical Study of Geographic User Activity Patterns in Foursquare. In: Int. Conf. on ICWSM (2011)
4. Ye, M., Yin, P., Lee, W.C., Lee, D.L.: Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation. In: Int. Conf. on SIGIR, pp. 325–334 (2011)
5. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Int. Conf. on WWW, pp. 791–800 (2009)
6. Domingos, P., Richardson, M.: Mining the Network Value of Customers. In: Int. Conf. on KDD, pp. 57–66 (2001)
7. Kempe, D., Kleinberg, J.M., Tardos, É.: Influential Nodes in a Diffusion Model for Social Networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
8. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the Spread of Influence through a Social Network. In: Int. Conf. on KDD, pp. 137–146 (2003)
9. Granovetter, M.: Threshold Models of Collective Behavior. American Journal of Sociology 83, 1420–1443 (1978)
10. Vasconcelos, M.A., Ricci, S.M.R., Almeida, J.M., Benevenuto, F., Almeida, V.A.F.: Tips, Dones and Todos: Uncovering User Profiles in Foursquare. In: Int. Conf. on WSDM, pp. 653–662 (2012)