

RESEARCH ARTICLE

# Recurrent spatio-temporal modeling of check-ins in location-based social networks

Ali Zarezade, Sina Jafarzadeh, Hamid R. Rabiee\*

ICT Innovation Center, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

\* [rabiee@sharif.edu](mailto:rabiee@sharif.edu)



## Abstract

Social networks are getting closer to our real physical world. People share the exact location and time of their check-ins and are influenced by their friends. Modeling the spatio-temporal behavior of users in social networks is of great importance for predicting the future behavior of users, controlling the users' movements, and finding the latent influence network. It is observed that users have periodic patterns in their movements. Also, they are influenced by the locations that their close friends recently visited. Leveraging these two observations, we propose a probabilistic model based on a doubly stochastic point process with a periodic-decaying kernel for the time of check-ins and a time-varying multinomial distribution for the location of check-ins of users in the location-based social networks. We learn the model parameters by using an efficient EM algorithm, which distributes over the users, and has a linear time complexity. Experiments on synthetic and real data gathered from Foursquare show that the proposed inference algorithm learns the parameters efficiently and our method models the real data better than other alternatives.

## OPEN ACCESS

**Citation:** Zarezade A, Jafarzadeh S, Rabiee HR (2018) Recurrent spatio-temporal modeling of check-ins in location-based social networks. PLOS ONE 13(5): e0197683. <https://doi.org/10.1371/journal.pone.0197683>

**Editor:** Dejing Dou, University of Oregon, UNITED STATES

**Received:** November 23, 2017

**Accepted:** May 7, 2018

**Published:** May 23, 2018

**Copyright:** © 2018 Zarezade et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All datasets are available through our git repository at <https://github.com/azarezade/STP/tree/master/code/Data>. Our data collection method complies with the terms of service of both Twitter and Foursquare. Moreover, the datasets are anonymized and does not reveal the identity of actual users.

**Funding:** The author(s) received no specific funding for this work.

**Competing interests:** The authors have declared that no competing interests exist.

## Introduction

The advances in location-acquisition techniques and the proliferation of mobile devices have generated an enormous amount of spatial and temporal data of users activities [1]. People can upload a geotagged video, photo or text to social networks like Facebook and Twitter, share their present location on Foursquare or share their travel route using GPS trajectories to Geo-Life [2]. A considerable amount of spatio-temporal data is generated by the activity of users in location-based social networks (LBSN). In a typical LBSN, like Foursquare, users share the time and geolocation of their check-ins, comment about a venue, or unlock badges by exploring new venues. These data motivated the researchers to study the human spatio-temporal behavior in social networks [3, 4].

Many techniques have been proposed for processing, managing, and mining the trajectory data in the past decade [5]. Several other studies try to leverage the spatial data in recommender systems [6]. However, a few works have attempted to model the recurrent spatio-temporal behavior of users in LBSNs [7, 8]. Given the history of users' check-ins, the goal is to predict the time and location of users' check-ins utilizing a model. This model can also be used to find the influence network between users which made up of their check-ins, detect the

influential users and popular locations, predict the peak hours of a restaurant, recommend a location, and even control the movement of users.

In this paper, we propose a probabilistic generative model for the check-ins of users in location-based social networks, which can be used in predicting the future check-ins of the users, and discovering the latent influence network. People usually have periodic patterns in their movements [8–10]. For example, a typical user may check into her office in the morning and to a nearby restaurant at noon then return home and repeat this behavior in the following days. We model the time of check-ins of each user with a novel periodic-decaying doubly stochastic point process which leverages the periodicity in the movements of users and can also capture any drift in their patterns. To model the location of check-ins we use the fact that users in social media are influenced by the activities of their friends [11–13]. If many of your close friends have checked into a specific restaurant recently, then there is a high probability that you select that restaurant, next time. We model the location of check-ins using a time-varying multinomial distribution. In summary, we propose:

- Doubly stochastic point process for modeling the time of users' check-ins, which captures the periodic behavior in the movement of users.
- Time-varying multinomial distribution for modeling the location of users' check-ins, which incorporates the mutually-exciting effect of the friends' history.
- Scalable inference algorithm based on the EM algorithm to find the model parameters, which is distributed over users, and has a linear time complexity.
- Compelling dataset of Foursquare users' check-ins, curated from 12000 active users during three months in the year 2015.

Our work relates to previous work on temporal point processes, and location-based social networks analysis.

Modeling information diffusion in social networks has attracted a lot of attentions in recent years. Given the times that users have adopted to a contagion (information, behavior, or meme), the problem is to model the time and user of the next adoption, *i.e.*, predict the next event. Early methods [14, 15] studied information diffusion using a pair-wise probability distribution for each link from node  $j$  to  $i$ , which is the probability that node  $i$  generates an event in time  $t_i$  due to the event of node  $j$  at time  $t_j$ . These methods overlook the external effects on the generation of events. In addition, they assume that each node adopts a contagion at most once, *i.e.*, events are not recurrent. These issues were later addressed in [7, 16–20], which they use point processes for the modeling of events. In [15–17, 19], cascades are assumed to be independent and are modeled by a special point process, called Hawkes [21]. The independence assumption is removed in [11, 22], they tried to model the correlation between multiple competing or cooperating cascades. In [7] a spatio-temporal model is proposed for the interactions between a pair of users not an individual user as in our model. Other studies [17, 23–26], use the additional information of the diffusion network such as topic of tweets or the community structure to better model the influence network. Moreover, in [27, 28], a stochastic optimal control framework is proposed to control the diffusion process in complex networks. Recently, the recurrent neural networks (RNN) are utilized to learn the intensity function of a temporal point process as a nonlinear function of the history and solve the resulting nonlinear optimization by a stochastic gradient algorithm [29, 30]. Most of the previous works studied the temporal diffusion of information on microblogging networks like Twitter, whereas we try to model the time and location of users' check-ins in the location-based networks like Foursquare. Moreover we proposed a periodic point process which is of independent importance,

whereas in the previous studies the self-exciting point processes is used for the modeling of events.

The prior works in location-based social networks can be categorized into three groups [6]: location recommendation, trajectory mining and location prediction. The main approaches in location recommendation systems are: content-based which uses data from a user’s profile and the features of locations [31, 32]; link-based, which applies link analysis models like PageRank to identify the experienced users and interesting locations [33, 34]; and collaborative filtering which infers users’ preferences from their historical behavior, like the location history [35, 36]. In trajectory data mining, the source of data is usually generated by the GPS. These works include; trajectory pattern mining to find the next location of an individual [8, 37–39], anomaly detection to detect unexpected movement patterns [40, 41], and trajectory classification to differentiate between trajectories of different states, such as motions, transportation modes, and human activities [42]. A comprehensive review of these methods can be found in the recent survey [5]. We also discriminate our work from location recommendation and trajectory mining methods, because our goal is to model the check-ins of users not to recommend a location or to find the trajectory patterns of users with the position data of their routes. In location prediction, the goal is to predict the next location, given the user’s profile data and the history of check-ins. But these methods do not consider; the relation between friends (using the influence matrix), aging effect in the history of checkins (using decaying kernel), exogenous effects on users’ decisions, and periodicity in users’ movement patterns.

## Materials and methods

### Preliminaries

To model the time of occurrences of a phenomenon, which are called events, we can use point processes on the real line. The phenomena can be, an earthquake [43], a viral disease [44] or the spread of information over a network [15]. The sequence of events, as defined below, is the realization of a point process.

**Definition 1** (Point Process). *Let  $\{t_i\}_{i \in \mathbb{N}}$  be a sequence of non-negative random variables such that  $\forall i \in \mathbb{N}, t_i < t_{i+1}$ , then we call  $\{t_i\}_{i \in \mathbb{N}}$  a point process on  $\mathbb{R}$ , and  $\mathcal{F}_t = \{t_i \mid i \in \mathbb{N}, t_i < t\}$  as its history or filtration.*

There are different equivalent descriptions for the point processes such as; sequence of points  $\{t_i\}$ , sequence of intervals (duration process)  $\delta t_i$ , counting process  $N(t)$ , or intensity process  $\lambda(t)$  [45]. In the following, we briefly explain each definition.

The counting process  $N(t)$  associated with the point process  $\{t_i\}_{i \in \mathbb{N}}$ , counts the number of events occurred before time  $t$ , i.e.,  $N(t) = \sum_{i \in \mathbb{N}} \mathbb{I}(t_i < t)$ , where indicator function  $\mathbb{I}(x \in A)$  is 1 if  $x \in A$ , and is 0 otherwise. The duration process  $\delta t_i$  associated with the point process  $\{t_i\}_{i \in \mathbb{N}}$  is defined as  $\forall i \in \mathbb{N}, \delta t_i = t_i - t_{i-1}$ . Finally, the intensity process  $\lambda(t)$  is defined as the expected number of events per units of time, which generally depends on the history:

$$\begin{aligned} \lambda(t|\mathcal{F}_t) &= \lim_{dt \rightarrow 0} \frac{1}{dt} \mathbb{E}[N(t, t + dt) \mid \mathcal{F}_t] \\ &= \lim_{dt \rightarrow 0} \frac{1}{dt} \Pr[N(t, t + dt) > 0 \mid \mathcal{F}_t] \end{aligned}$$

where  $N(t, s] := N(s) - N(t)$ . To evaluate the likelihood of a sequence of events,  $f(t_1, t_2, \dots, t_n)$ , we can use the chain rule of probability,  $f(t_1, t_2, \dots, t_n) = \prod_i f(t_i | t_{1:i-1})$ . Therefore, it suffice to describe only the conditionals, which are abbreviated to  $f^*(t)$ . According to the definition of

point processes, we can write the probability of occurring the  $(n + 1)$ 'th event in time  $t$  as:

$$f^*(t) dt = \Pr\{N(t_n, t] = 0, N(t, t + dt] = 1 \mid \mathcal{F}_{1:n}\}.$$

If we divide both sides of the above equation by  $1 - F^*(t)$ , where  $F^*(\cdot)$  is the cdf of  $f^*(\cdot)$ , then in the limit as  $dt \rightarrow 0$ , we have:

$$\begin{aligned} \frac{f^*(t) dt}{1 - F^*(t)} &= \frac{\Pr\{N(t_n, t] = 0, N(t, t + dt] = 1 \mid \mathcal{F}_{1:n}\}}{\Pr\{N(t_n, t] = 0 \mid \mathcal{F}_{1:n}\}} \\ &= \Pr\{N(t, t + dt] = 1 \mid \mathcal{F}_{1:n}, N(t_n, t] = 0\} \\ &= \Pr\{N(t, t + dt] > 0 \mid \mathcal{F}_t\} \end{aligned}$$

Therefore, according to the definition of intensity, we find the relation between conditional distribution of the time of events and the intensity function as:

$$\lambda^*(t) = \frac{f^*(t)}{1 - F^*(t)} \tag{1}$$

where we use \* superscript to show that a function is dependent on the history. We can also express the relation of  $\lambda^*(t)$  and  $f^*(t)$  in the reverse direction [46]:

$$f^*(t) = \lambda^*(t) \exp\left(-\int_{t_n}^t \lambda^*(s) ds\right) \tag{2}$$

Now, the cdf can be easily evaluated:

$$F^*(t) = 1 - \exp\left(-\int_{t_n}^t \lambda^*(s) ds\right). \tag{3}$$

A point process is usually defined by specifying its conditional distribution  $f^*(t)$  or equivalently its intensity  $\lambda^*(t)$ . In the simplest case, the intervals  $\delta t_i$  are assumed to be *i.i.d.*, therefore the process is memoryless, and hence  $\lambda^*(t) = \lambda(t)$ . The Cox process [47] is a doubly stochastic point processes, and conditioned on the intensity is a Poisson process [48]. Hawkes process [21] is a special type of Cox process, where the intensity is expressed by the history as:

$$\lambda^*(t) = \mu + \int_{-\infty}^t \phi(t - \tau) dN(\tau) = \mu + \sum_{i=1}^{|\mathcal{F}_t|} \phi(t - t_i) \tag{4}$$

where  $\phi(t)$  is the kernel of the Hawkes process that defines the effect of past events on the current intensity, and  $\mu$  is the base intensity. For example, the exponential kernel  $\phi(t) = \exp(-t)$ , is used for the modeling of self-exciting events like earthquake [43]. In general, we have a multivariate process with a counting process vector  $\mathbf{N}(t) = [N_1(t), \dots, N_n(t)]^T$  and an associated intensity vector  $\boldsymbol{\lambda}^*(t) = [\lambda_1^*(t), \dots, \lambda_n^*(t)]^T$  defined as:

$$\boldsymbol{\lambda}^*(t) = \boldsymbol{\mu} + \mathbf{A} \int_{-\infty}^t \boldsymbol{\Phi}(t - \tau) d\mathbf{N}(\tau) \tag{5}$$

where  $\boldsymbol{\Phi}(t)$  is the matrix of mutual kernels, *i.e.*,  $\Phi_{ij}(t)$  models the effect of events of counting process  $N_j(t)$  on  $N_i(t)$ ,  $\boldsymbol{\mu} = [\mu_1, \dots, \mu_n]^T$  is the base intensity, and  $\mathbf{A} = [\alpha_{ij}]$  is a matrix of mutual-excitation kernels. Often, the point process carries other information than the time of events, which is called mark. For example, the strength of an earthquake can be considered as a mark. The mark  $m$ , often a subset of  $\mathbb{N}$  or  $\mathbb{R}$ , is associated with each event through the conditional

mark probability function  $f^*(m|t)$ :

$$\lambda^*(t, m) = \lambda^*(t) f^*(m|t) \tag{6}$$

The mutually-exciting property of the Hawkes process makes it a common modeling tool in a variety of applications such as seismology, neurophysiology, epidemiology, reliability, and social network analysis [14, 15, 22].

### Problem definition

Given the history of users activity in a location based social network,  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , with  $|\mathcal{V}| = N$  users and  $L$  locations in  $C$  different categories, we propose a generative model for the check-ins of users. In other words, for each user we can predict the location and time of her next check-in.

We define a check-in as a 4-tuple  $(t, u, c, l)$ , which shows the time  $t$  that user  $u$  check-in to location  $l$  with category  $c$ . We observe the sequence of all check-ins in the network  $\mathcal{G}$ , in the time interval  $[0, T]$ . The observation  $\mathcal{D} = \{(t_i, u_i, c_i, l_i)\}_{i=1}^K$  is composed of user's check-ins where  $t_i \in [0, T]$ ,  $u_i \in \mathcal{V}$ ,  $c_i \in \{1, 2, \dots, C\}$  and  $l_i \in \{\phi_1, \phi_2, \dots, \phi_L\}$ , where  $\phi_i$  is the unique id of the  $i$ 'th location. Since we use location ids instead of geo-coordinates, it is fair to assume the observation data is noiseless, however in practice, there may be an uncertainty in the locations of check-ins, which we are considering it as a future work. We use the following notation for the history of check-ins of user  $u$  in location  $l$  with category  $c$  up to time  $t$ :

$$\mathcal{D}_{ucl}(t) = \{(t_i, u_i, c_i, l_i) \in \mathcal{D} \mid t_i < t, u_i = u, c_i = c, l_i = \phi_l\}$$

Moreover, we use the dot notation to represent the union over the dotted variable, e.g.,  $\mathcal{D}_{u..}(t)$  represents the events of user  $u$ , before time  $t$ , in any location with any category, and  $\mathcal{D}_{..c}(t)$  represents the events of all users except  $u$ , before the time  $t$ , in any location with category  $c$ .

By observing the periodic pattern in the time of users' checkins (see the Results section) we model the time of check-ins using a doubly stochastic point process which incorporates both the periodic patterns and exogenous effects, in the users' movements. The exogenous effects are any other external effects on the users' time of check-ins which are not necessarily periodic. To model the location of check-ins we propose a time-dependent multinomial distribution which incorporates the mutually-exciting effect of friends, which this effect is also empirically observed in the real data.

### Proposed method

**modeling the time of check-ins.** In every working day, a user may check-in to her office in the morning then go to a restaurant at noon, and also have a weekly soccer practice program. By observing the history of the time of check-ins of a user, if she repeats some patterns recently (within several days), for example take a walk every afternoon, then it is more likely to repeat this pattern shortly in the upcoming days at approximately the same time. It means, there is a periodicity in the users' behaviors. Moreover, there maybe also a drift or an addition of a new activity in the user's behavior, for example, the working hour of her office may change or there may be a new weekly social gathering. Therefore, we need a periodic point process to model the time of user's check-ins, which can also adapt to the new users' check-ins. This is in contrast to the self-exciting nature of the Hawkes process, which is used to model the diffusion of information over a network [14, 15, 17].

We propose a doubly stochastic point process which is periodic, and also has a diminishing property that enables the process to change its periodic pattern and adapt to the new behaviors. The proposed process, is composed of a Poisson process with the base intensity  $\mu$ , where each

event  $t_i$  of this process triggers a Poisson process with the following intensity:

$$\lambda_{t_i}(t) = \sum_{k=1}^{\infty} h(t - t_i - k\tau) g(k) \tag{7}$$

where  $h(t)$  is the kernel of the process,  $g(k)$  is a decreasing function to diminish the intensity in the future periods, and the hyper-parameter  $\tau$  is the period. This intensity is illustrated in Fig 1. The self-exciting property of the Hawkes process can be observed from its exponentially decaying kernel in Fig 1. In the Hawkes process when an event occurs, there is a high probability to have events just after it, and this probability decreases exponentially afterward. But in the proposed process, there is a high probability to have events in the upcoming periods and this probability also decreases exponentially.

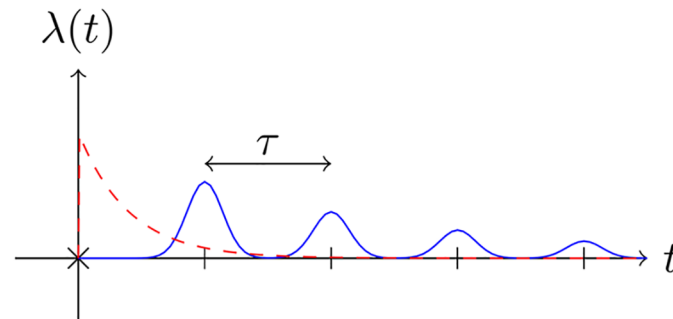
According to the superposition theorem [48], the intensity of the proposed process can be written as follows:

$$\lambda^*(t) = \mu + \sum_{i=1}^{|\mathcal{F}_t|} \lambda_{t_i}(t) = \mu + \sum_{i=1}^{|\mathcal{F}_t|} \sum_{k=1}^{\infty} h(t - t_i - k\tau) g(k) \tag{8}$$

To preserve the locality in time, the kernel  $h(t)$  should have a peak at  $t = 0$  and decay to zero in both sides when  $t \rightarrow \pm\infty$ . For example, the Gaussian kernel,  $h(t) = \exp(-t^2/2\sigma^2)$  meets this requirements. This model has three main features:

1. *Periodic Nature.* When an event occurs in time  $s$ , then the intensity of events around this time in the upcoming periods,  $s + k\tau$ , would increase.
2. *Temporal Locality.* The intensity is high around the peak of the kernel and drops rapidly in both sides.
3. *Adaptability.* The peak of the kernel decreases by the increase of  $k$ , so the process can adopt its intensity to any new periodic patterns.
4. *Exogenous Effect.* Other external effects can be modeled by the base intensity  $\mu$ .

If we use a truncated Gaussian kernel like  $h(t) = \exp(-t^2/2\sigma^2) \mathbb{I}(-\tau/2 \leq t \leq \tau/2)$ , then we can substantially reduce the complexity of the intensity function. With this kernel we can



**Fig 1. Periodic point process.** An event at time  $t = 0$  triggers a poisson process. The solid curve shows the intensity of the proposed periodic point process with a Gaussian kernel and period  $\tau$ , and the dashed curve shows a Hawkes process with an exponential decaying kernel.

<https://doi.org/10.1371/journal.pone.0197683.g001>

show that:

$$\lambda^*(t) = \mu + \sum_{i=1}^{|\mathcal{F}_t|} h(t - t_i - k_i\tau) g(k_i) \tag{9}$$

where  $k_i = \lfloor \frac{t-t_i}{\tau} \rfloor$  is the period number of which the event in  $t_i$  affects on the current intensity. So, we propose the following point process for the time of check-ins of user  $u$  in any location with category  $c$ :

$$\lambda_u(t, c) = \mu_{uc} + \sum_{i=1}^{|\mathcal{D}_{uc}(t)|} \beta_u \exp \left[ -\frac{(t - t_i - k_i\tau)^2}{2\sigma^2} \right] \exp(-k_i) \tag{10}$$

The first term,  $\mu_{uc}$  is the base intensity that models the external effect on user  $u$  to generates check-ins with category  $c$ , the second term is the periodic effect of the history,  $\beta_u$  is the kernel parameter, and  $\tau, \sigma$  are hyper-parameters. All parameters of the model are listed in Table 1. The intuition of this model is that, if a user check-ins frequently, for example in the “restaurant” category at noon, then with high probability, she will checks in a restaurant at noon in the next day.

**modeling the location of check-ins.** In this section, we propose a model for the location of users’ check-ins, given the history of check-ins. We use the fact that, users in social networks are influenced by the behavior of their neighbors. Let denote the weight of location  $l$  with category  $c$  for user  $u$  as:

$$w_{ucl} = \sum_{i=1}^{|\mathcal{D}_{cl}(t)|} \alpha_{u_i, u} \exp(-(t - t_i)) \tag{11}$$

which incorporates  $\alpha_{u_i, u}$ , the influence of user  $u_i$  on  $u$ , and the time of check-ins with an exponentially decaying kernel. This kernel diminishes the effect of far past check-ins, so the model can adopt to any new behaviors of the users’ check-ins. Therefore, a location which checked in recently with many or even few but influential friends would have high weight. We also define a weight for the popularity of a location  $l$  with category  $c$  from the perspective of all users:

$$m_{cl} = \sum_{i=1}^{|\mathcal{D}_{cl}(t)|} \exp(-(t - t_i)) \tag{12}$$

where the location that is most checked in recently, has the highest weight.

When a user decides to check-in for example, at a restaurant, she selects a location that herself or her friends have checked in frequently, recently (exploitation effect), and sometimes she check-ins to a new popular restaurant (exploration effect). Therefore, we use the following multinomial conditional distribution to define the probability that user  $u$  check-ins to location

**Table 1. Parameters of the model.**

Parameter	Description
$\beta_u$	Temporal kernel parameter of user $u$
$\mu_{uc}$	Base temporal intensity of user $u$ in category $c$
$\alpha_{vu}$	The influence of users $v$ on $u$
$\eta_{uc}$	Tendency of user $u$ to explores new locations with category $c$

<https://doi.org/10.1371/journal.pone.0197683.t001>



$\ell$ , given the time  $t$  and category  $c$ :

$$f_u(\ell|c, t) = \underbrace{\sum_{l=1}^L \frac{w_{ucl}}{\eta_{uc} + w_{uc}} \delta_{\phi_l}(\ell)}_{\text{exploitation}} + \underbrace{\frac{\eta_{uc}}{\eta_{uc} + w_{uc}} G_0(\ell)}_{\text{exploration}} \tag{13}$$

The Dirac delta function  $\delta_{\phi_l}(\ell)$  is 1 if  $\phi_l = \ell$ , otherwise it is 0, and the parameter  $\eta_{uc}$  models the inclination of the user to explore new locations. This distribution means that, with probability  $w_{ucl}/(\eta_{uc} + w_{uc})$  the current location would be a previously checked in location  $\phi_l$  by the user  $u$  or any of her friends (since for non visited locations the weight  $w_{ucl}$  is zero), and with probability  $\eta_{uc}/(\eta_{uc} + w_{uc})$  it would be selected from all locations in the network, with a probability that is modeled by the following distribution:

$$G_0(\ell) = \sum_{l=1}^L \frac{m_l}{m_c} \delta_{\phi_l}(\ell) \tag{14}$$

Where according to the definition of coefficient  $m_{cb}$  it assigns more probability to the popular or recently frequently visited locations. The main features of the proposed location model are:

1. *Exploitation.* The future check-ins of a user are influenced by the history of check-ins of the user and her friends.
2. *Exploration.* There is a probability that users explore and check into new unseen locations.
3. *Adaptability.* Using exponential decaying kernel for the weights, the model can adopt to new patterns in users' behavior.
4. *Influence Network.* The parameters  $\{\alpha_{vu}\}$  are actually modeling the latent influence network which are learned from the check-ins history.

**summary of the generative model.** The proposed generative model is summarized in Alg. 1. Using the superposition theorem, first the time  $t$  of check-in is sampled from the proposed periodic point process  $\lambda(t) = \sum_{u,c} \lambda_u(t, c)$ , then the user  $u$  which generated this event is selected in proportion to its intensity  $\lambda_u(t)$ . The category  $c$  of the check-in is also selected in proportion to  $\lambda_u(t, c)$ . Finally, the location  $l$  is sampled from the proposed location model.

**Algorithm 1:** Generative model of the check-ins.

```

Input:  $N, C, L$ , all parameters  $\{\mu_{uc}, \eta_{uc}, \alpha_{uv}, \beta_u\}$ , history of check-ins.
Output: Next check-in  $(t_i, u_i, c_i, l_i)$ .
for  $u = 1 : N$  do
     $\lambda_u(t) = \sum_c \lambda_u(t, c)$ 
end
 $\lambda(t) = \sum_u \lambda_u(t)$ 
 $t_i \sim \mathcal{PP}(\lambda(t))$ 
 $u_i \sim \text{Multi}\left(\frac{\lambda_1(t_i)}{\lambda(t_i)}, \dots, \frac{\lambda_N(t_i)}{\lambda(t_i)}\right)$ 
 $c_i \sim \text{Multi}\left(\frac{\lambda_{u_i}(t_i, 1)}{\lambda_{u_i}(t_i)}, \dots, \frac{\lambda_{u_i}(t_i, C)}{\lambda_{u_i}(t_i)}\right)$ 
 $l_i \sim \mathcal{F}_{u_i}(\ell | c_i, t_i)$ 
return  $(t_i, u_i, c_i, l_i)$ 

```

**inference.** We propose a Bayesian inference algorithm based on the EM algorithm to find the model parameters. To find the maximum likelihood solution, for each check-in  $(t_i, u_i, c_i, l_i)$ , we define a latent variable  $z_i$  as the user that caused  $u_i$  to check into location  $l_i$ , given the time  $t_i$  and category  $c_i$ . We use 1-of- $N$  coding to represent  $z_i$ 's. For notional convenient, lets



define:

$$\gamma_{ucl}^v = \frac{w_{ucl}^v}{\eta_{uc} + w_{uc}} \mathbb{I}(v > 0) + \frac{m_{cl} \eta_{uc}}{m_c (\eta_{uc} + w_{uc})} \mathbb{I}(v = 0) \tag{15}$$

$$w_{ucl}^v = \sum_{i=1}^{|\mathcal{D}_{vcl}(t)|} \alpha_{vu} \exp(-(t - t_i)) = \alpha_{vu} \sum_{i=1}^{|\mathcal{D}_{vcl}(t)|} \exp(-(t - t_i)) \tag{16}$$

where  $\gamma_{ucl}^v$  is the contribution or influence of user  $v$  in the check-in of user  $u$  at location  $l$  with category  $c$ . Now, we define:

$$f_{u_i}(l_i, z_i | t_i, c_i) = \prod_{v=0}^N (\gamma_{u_i c_i l_i}^v)^{z_{iv}} \tag{17}$$

where  $z_{iv}$  is the  $v$ 'th element of  $z_i$ , or the index of the user that caused  $i$ 'th check-ins. But,  $v = 0$  is not the index of a user, it represents the exploration effect. It can be verified that marginalizing out the  $z_i$ ,  $\sum_{z_i} f_{u_i}(l_i, z_i | t_i, c_i)$ , results in the probability distribution (13). Now, to evaluate the complete likelihood  $p(\mathcal{D}, Z | \theta)$  of the data  $\mathcal{D}$  and hidden variables  $Z = \{z_i\}_{i=1}^K$ , given the parameters  $\theta = \{\mu_{uc}, \eta_{uc}, \alpha_{uv}, \beta_u\}$ ,  $u = 1 \dots N$ ,  $v \in \mathcal{N}(u)$  and  $c = 1 \dots C$ , where  $\mathcal{N}(u)$  is the set of neighbors of  $u$ , we use the following proposition.

**Proposition 1 ([11])** Let  $N_u$ ,  $u = 1, 2, \dots, N$  be a multivariate marked point process with the associated intensity  $\lambda_u(t)$ , and the mark probability  $f_u(m|t)$ . Let  $\mathcal{D} = \{(t_i, u_i, m_i)\}_{i=1}^K$  be a realization of the process over  $[0, T]$ . Then the likelihood of  $\mathcal{D}$  on model  $N_u$  with parameters  $\theta$  can be expressed as follows.

$$p(\mathcal{D} | \theta) = \exp\left(-\int_0^T \sum_{u=1}^N \lambda_u(\tau) d\tau\right) \prod_{i=1}^{|\mathcal{D}|} \lambda_{u_i}(t_i) f_{u_i}(m_i | t_i)$$

If we consider  $(c_i, l_i, z_i)$  as the mark  $m_i$  of the process, according to this proposition the complete likelihood of our model is,

$$p(\mathcal{D}, Z | \theta) = \exp\left(-\int_0^T \sum_{u=1}^N \lambda_u(\tau) d\tau\right) \prod_{i=1}^{|\mathcal{D}|} \lambda_{u_i}(t_i) f_{u_i}(c_i, l_i, z_i | t_i) \tag{18}$$

where using Bayes' rule and Eq (17) it can be evaluated as follows.

$$\begin{aligned} p(\mathcal{D}, Z | \theta) &= \exp\left(-\int_0^T \sum_{u=1}^N \lambda_u(\tau) d\tau\right) \prod_{i=1}^{|\mathcal{D}|} \lambda_{u_i}(t_i) f_{u_i}(c_i | t_i) f_{u_i}(l_i, z_i | t_i, c_i) \\ &= \exp\left(-\sum_{u=1}^N \sum_{c=1}^C \int_0^T \lambda_u(\tau, c) d\tau\right) \prod_{i=1}^{|\mathcal{D}|} \lambda_{u_i}(t_i, c_i) f_{u_i}(l_i, z_i | t_i, c_i) \\ &= \exp\left(-\sum_{u=1}^N \sum_{c=1}^C \int_0^T \lambda_u(\tau, c) d\tau\right) \prod_{i=1}^{|\mathcal{D}|} \lambda_{u_i}(t_i, c_i) \prod_{v=0}^N (\gamma_{u_i c_i l_i}^v)^{z_{iv}} \end{aligned}$$

To derive the second line, we used the superposition theorem, and the fact that the probability of a category, according to our generative model is  $f_{u_i}(c_i | t_i) = \lambda_{u_i}(t_i, c_i) / \lambda_{u_i}(t_i)$ . Given the joint distribution of the observed and latent variables  $p(\mathcal{D}, Z | \theta)$ , we use EM algorithm to maximize the likelihood function  $p(\mathcal{D} | \theta)$  with respect to  $\theta$ . In the E-step we evaluate  $p(Z | \mathcal{D}, \theta)$ . Using

Bayes' rule we can write the posterior distribution of the latent variables as,

$$p(Z|\mathcal{D}, \theta) \propto \prod_{i=1}^{|\mathcal{D}|} \prod_{v=0}^N (\gamma_{u_i c_i \ell_i}^v)^{z_{iv}} \tag{19}$$

which factorizes over  $i$ , so that  $z_i$ 's are independent with multinomial distribution and we can write the expected of  $z_{iv}$  under this distribution as follows.

$$\mathbb{E}[z_{iv}] = \frac{\sum_{z_{iv}} z_{iv} (\gamma_{u_i c_i \ell_i}^v)^{z_{iv}}}{\sum_{z_i} \prod_{v=0}^N (\gamma_{u_i c_i \ell_i}^v)^{z_{iv}}} = \frac{\gamma_{u_i c_i \ell_i}^v}{\sum_{v=0}^N \gamma_{u_i c_i \ell_i}^v} \tag{20}$$

In the M-step we maximize  $\mathbb{E}_Z[\ln p(\mathcal{D}, Z|\theta)]$  the expected complete log-likelihood, which can be decomposed to the sum of expected log-likelihoods of users  $\mathbb{E}_{Z_u}[\ln p(\mathcal{D}_u, Z_u|\theta_u)]$ .

$$\begin{aligned} \mathbb{E}_Z[\ln p(\mathcal{D}, Z|\theta)] &= -\sum_{u=1}^N \sum_{c=1}^C \int_0^T \lambda_u(\tau, c) d\tau + \sum_{i=1}^{|\mathcal{D}|} \log \lambda_{u_i}(t_i, c_i) + \sum_{i=1}^{|\mathcal{D}|} \sum_{v=0}^N \mathbb{E}[z_{iv}] \log \gamma_{u_i c_i \ell_i}^v \\ &= \sum_u \left( -\int_0^T \sum_{c=1}^C \lambda_u(\tau, c) d\tau + \sum_{i=1}^{|\mathcal{D}_u|} \log \lambda_{u_i}(t_i, c_i) + \sum_{i=1}^{|\mathcal{D}_u|} \sum_{v=0}^N \mathbb{E}[z_{iv}] \log \gamma_{u_i c_i \ell_i}^v \right) \\ &= \sum_u \mathbb{E}_{Z_u}[\ln p(\mathcal{D}_u, Z_u|\theta_u)] \end{aligned} \tag{21}$$

Where  $Z_u = \{z_i \in Z \mid u_i = u\}$  and  $\theta_u = \{\mu_{uc}, \eta_{uc}, \alpha_{uv}, \beta_u\}$ ,  $v \in \mathcal{N}(u)$ ,  $c = 1 \cdot \cdot \cdot C$ . Accordingly, the M-step can be decomposed to multiple maximizations over users, which can be done in parallel. Therefore, for each user  $u$ , the two steps of the EM algorithm can be summarized as follows.

$$\text{E-Step: } \mathbb{E}[z_{iv}] = \frac{\gamma_{u_i c_i \ell_i}^v}{\sum_{v=0}^N \gamma_{u_i c_i \ell_i}^v} \tag{22}$$

$$\text{M-Step: } \theta_u^* = \arg \max_{\theta_u \geq 0} \mathbb{E}_{Z_u}[\ln p(\mathcal{D}_u, Z_u|\theta_u)] \tag{23}$$

In the following proposition, we prove that the maximization in M-step is concave, so it has a unique and optimal solution. Moreover, the performance of the overall inference algorithm is not affected by the network size, as long as the average degree of the network and the average number of events per users remains fixed. Since, they define the number of parameters and observed data of each EM inference algorithm, and consequently define the performance of the overall inference algorithm.

**Proposition 2.** *The expected log-likelihood of a user,  $\mathbb{E}_{Z_u}[\ln p(\mathcal{D}_u, Z_u|\theta_u)]$  as a function of  $\{\mu_{uc}, \tilde{\eta}_{uc}, \tilde{\alpha}_{uv}, \beta_u\}$  is concave, where  $\alpha_{uv} = \exp(\tilde{\alpha}_{uv})$  and  $\eta_{uc} = \exp(\tilde{\eta}_{uc})$ .*

*Proof.* According to Eq (21) the log-likelihood of user  $u$  is:

$$\mathbb{E}_{Z_u}[\ln p(\mathcal{D}_u, Z_u|\theta_u)] = -\int_0^T \sum_{c=1}^C \lambda_u(\tau, c) d\tau + \sum_{i=1}^{|\mathcal{D}_u|} \log \lambda_{u_i}(t_i, c_i) + \sum_{i=1}^{|\mathcal{D}_u|} \sum_{v=0}^N \mathbb{E}[z_{iv}] \log \gamma_{u_i c_i \ell_i}^v$$

The first term is a linear function of  $\{\mu_{uc}, \beta_u\}$ , so it is both convex and concave. The second term is the log of a linear function which is concave, according to composition rules [49]. The

third term is composed of  $\log \gamma_{uc_i \ell_i}^v$ , which for  $v > 0$ ,

$$\log \gamma_{uc_i \ell_i}^v = \tilde{\alpha}_{vu} - \log \left( e^{\tilde{\eta}_{uc_i}} + \sum_{j=1}^{|\mathcal{D}_{c_i}(t)|} e^{\tilde{\alpha}_{uj\mu}} e^{-(t-t_j)} \right) + \text{const}$$

and for  $v = 0$ ,

$$\log \gamma_{uc_i \ell_i}^v = \tilde{\eta}_{uc_i} - \log \left( e^{\tilde{\eta}_{uc_i}} + \sum_{j=1}^{|\mathcal{D}_{c_i}(t)|} e^{\tilde{\alpha}_{uj\mu}} e^{-(t-t_j)} \right) + \text{const}.$$

In both cases  $\log \gamma_{uc_i \ell_i}^v$  is concave according to Lemma 1 of [11] which state that logarithm of sum of linear exponentials is convex. So, the overall expression is concave. Actually, we use  $\tilde{\eta}_{uc}$ ,  $\tilde{\alpha}_{uv}$  instead of  $\eta_{uc}$ ,  $\alpha_{uv}$  in the implementations, and solve the resulting concave optimization.

To find the time complexity of the inference algorithm, by carefully investigating all terms in the M-step, it can be verified that, each gradient descent iteration in maximization 23 has  $O(k_u h_u)$  operations, where  $k_u$ ,  $h_u$  are the number of neighbors and the size of history of user  $u$ , respectively. Therefore, the approximate order of the overall inference algorithm is  $O(mkhN)$ , where  $k$ ,  $h$  are the average network degree and events per user, and  $m$  is the number of EM iterations times the number of the gradient descent iterations. In practice, since  $m$  which depends on the desired tolerance in the EM algorithm, and  $k$  are constant (the average degree of the most real work networks are less than 10 [44]), the overall complexity can be simply expressed by  $O(hN)$ , which is linear with respect to the number of users  $N$ , and the average number of events per user  $h$ .

## Datasets

To evaluate the proposed method we use a synthetic data, and a real data gathered from users check-ins data in Foursquare. All dataset is available through our git repository, [github.com/azarezade/STP](https://github.com/azarezade/STP). Our data collection method complies with the terms of service of both Twitter and Foursquare. Moreover, the dataset is anonymized and does not reveal the identity of actual users.

## Results and discussion

In this section, using both synthetic and real data, we evaluate the performance of the proposed method. First, in the synthetic data experiments, we show that the proposed inference algorithm can learn the model parameters with high accuracy. Then in the real data experiments, we show that the proposed method outperform the other competing methods.

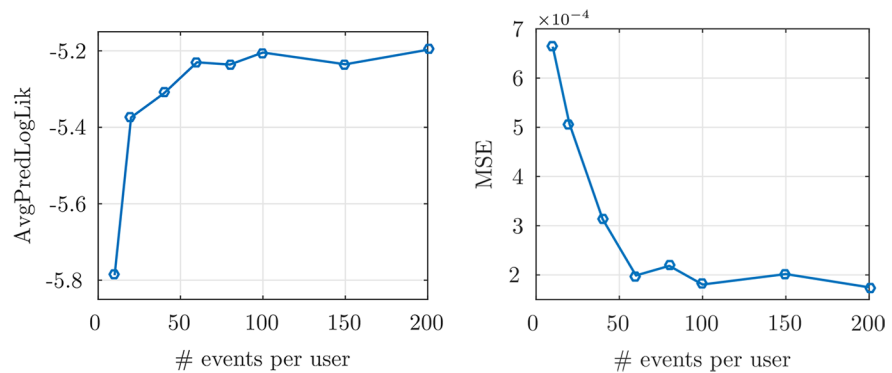
### Experiments on synthetic data

Following the literature, we use the synthetic data generated from our model to evaluate the performance of proposed learning algorithm. Moreover, we analyze the effect of model parameters on users behavior.

We experiment with five random Kronecker networks [50] with  $N = 64$  nodes, namely Core-periphery, Heterophily, Hierarchical, Homophily, and Erdos-Renyi, where the seed matrix parameters are [0.85, 0.45; 0.45, 0.3], [0.3, 0.89; 0.89, 0.3], [0.9, 0.1; 0.1, 0.9], [0.89, 0.3; 0.3, 0.89], and [0.60, 0.60; 0.60, 0.60], respectively. We set the number of categories to  $C = 4$  and consider eight locations in each category. The temporal and spatial model parameters are randomly drawn from the uniform distributions  $\mu_{uc}$ ,  $\eta_{uc} \sim U(0, 0.05)$ ,  $\alpha_{uv} \sim U(0, 0.5)$  and  $\beta_u$

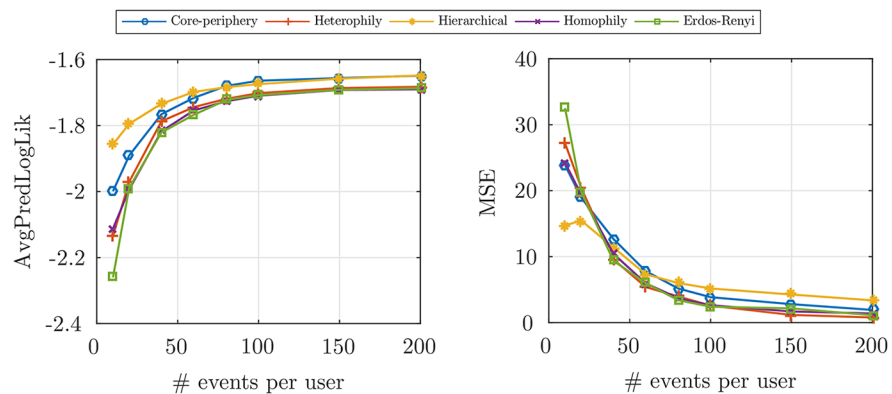
$\sim U(0, 0.1)$ . The period and standard deviation in the temporal model set to  $\tau = 12$  and  $\sigma = 0.5$ , respectively. We generate 16000 check-ins from our model, using the Ogata method [51], and consider the first 80% of them for the train and the remaining 20% for the test data. Then we learn the model with different percentages of the training data, and evaluate the average predicted log-likelihood on the test data (*AvgPredLogLik*) and the mean squared error between the estimated and real parameters (*MSE*). The inference algorithm is implemented in parallel for all users. All source codes and datasets are available in our git repository.

In Fig 2, the *AvgPredLogLik* and *MSE* of the temporal model is plotted versus the size of train data, where the average estimation error decreases to about  $7 \times 10^{-4}$ . These measures are also plotted for the spatial model with different random network structures in Fig 3, given the time of check-ins. We can see that the parameter estimation error decreases and the average log-likelihood increases as we increase the size of train data, which shows the proposed inference algorithm can consistently learn the model parameters with a very small estimation error.



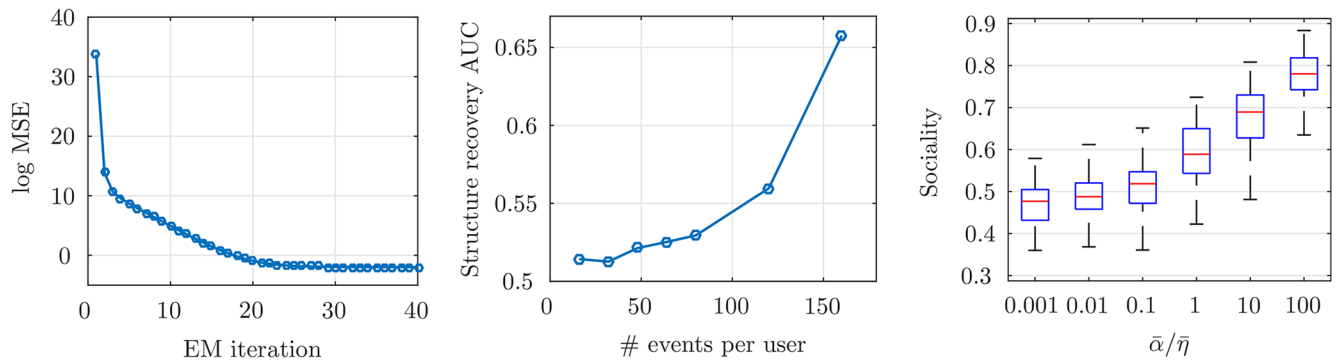
**Fig 2. Synthetic data temporal measures.** Average predicted log-likelihood on the test data (*left*), and MSE of the learned parameters (*right*), in the temporal model for the different percentages of the train data.

<https://doi.org/10.1371/journal.pone.0197683.g002>



**Fig 3. Synthetic data spatial measures.** Average predicted log-likelihood on the test data (*left*), and MSE of the learned parameters (*right*), in the spatial model for the different percentages of the train data and various random graph structures.

<https://doi.org/10.1371/journal.pone.0197683.g003>



**Fig 4. Synthetic data evaluations.** Average predicted log-likelihood in logarithmic scale vs the iterations of EM (left), the network structure recovery for different percentages of the train data (middle), and the effect of spatial parameters on the users' Sociality (right).

<https://doi.org/10.1371/journal.pone.0197683.g004>

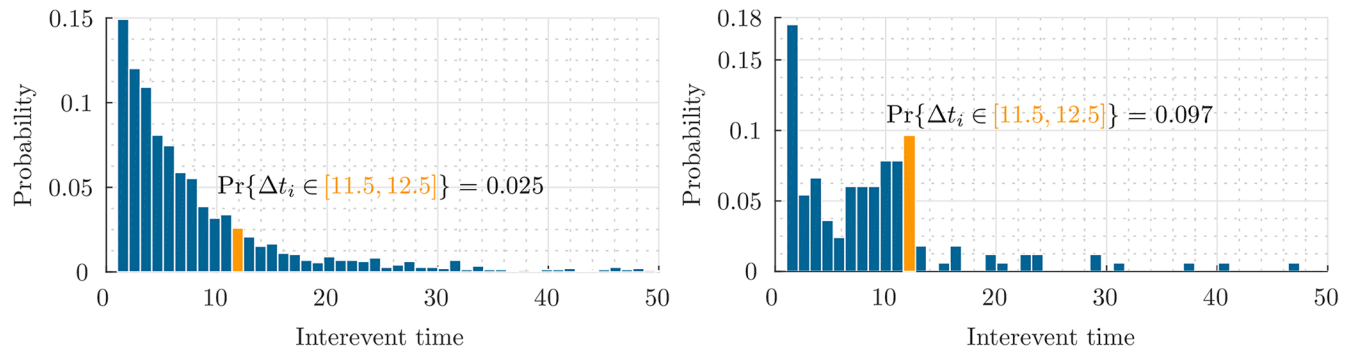
Furthermore, in the left of Fig 4, we show that for a fixed number of events per user, increasing the EM iterations would decrease MSE to about 0.1. To investigate the network structure prediction of our model, for each size of the train data, we use a threshold to convert the predicted weighted network (i.e., the  $\alpha_{ij}$ 's) to a (0, 1)-adjacency matrix and evaluate the percent of recovered edges to form the ROC curve. Then, we find the AUC curve, which is illustrated in the middle of Fig 4. Our method finds 64% of edges using only 150 events per user in the train data.

To study the effect of model parameters on the users' behavior, we design two experiments. First, we define a measure called *Sociality*. For each user, the *Sociality* is the percent of check-ins that their location has been previously visited by the user or her friends. According to our spatial model, Eq (13), the exploration of users increase as we increase  $\eta$  or decrease  $\alpha$ . To empirically validate this property of our model, in the right of Fig 4 the box plot of the users' *Sociality* is illustrated for different parameters. The average sociality reaches up to 80% when the average ratio of spatial parameters,  $\bar{\alpha}/\bar{\eta}$  is equal to 100. It means that, users with high  $\alpha/\eta$  are more affected by their friends. Moreover, to see the effect of temporal model parameters on the check-ins time of users, we plot the distribution of users' interevent time (the time difference between two successive events in a specific category for each user). According to Eq (10), parameters  $\beta$  and  $\mu$  regulate the periodicity in the time of events. The higher  $\beta$ , would result in more periodic events. We fix  $\mu$  and set  $\beta = 0$  and 1 in the left and right graphs of Fig 5, respectively. As we see, there is a peak around 12 in the right graph, which is the period of the simulated events but, in the left figure the frequency of events reduces exponentially and there is no peak except the initial one.

### Experiments on real data

In this section we use the real data gathered from users' checkins in Foursquare, which is a popular LBSN, to evaluated the proposed method against other alternative continuous time check-in models.

We used both Twitter and Foursquare APIs to crawl the check-ins data of the users in Foursquare, because Foursquare does not provide the full check-ins data. Specifically, we crawled the tweets of the users that have installed Swarm application, and publicly tweet their check-ins. This app is connected to the Twitter and Foursquare account of the user. When a user check-ins, using this app, she can tweet the URL of that location in the Foursquare website.



**Fig 5. Interevent distribution in Hawkes vs our periodic point process.** The distribution of interevent in the temporal model with  $\beta = 0$  (left) and  $\beta = 1$  (right). We can see that increasing  $\beta$  would cause a peak around 12, which is the period of the simulated events.

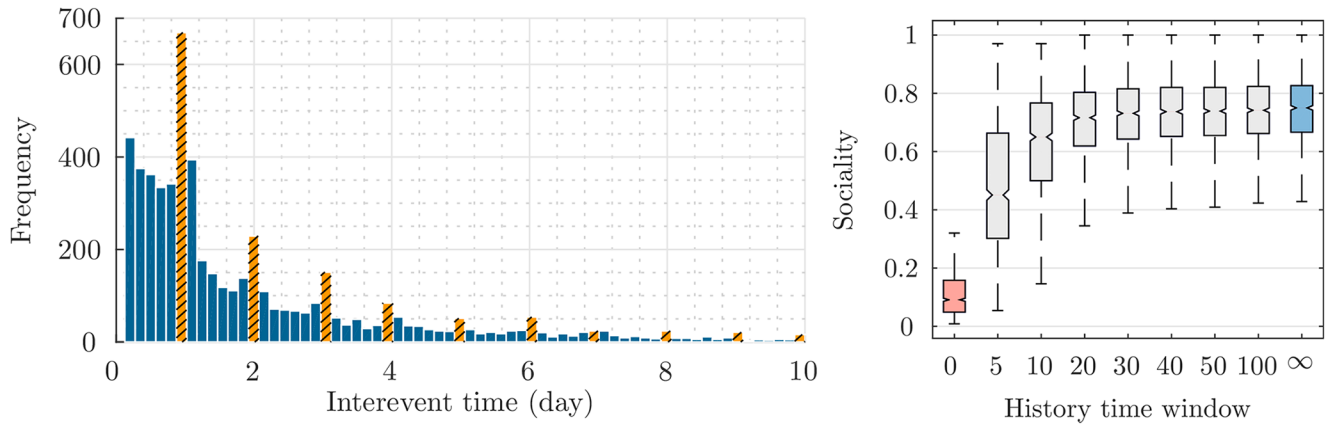
<https://doi.org/10.1371/journal.pone.0197683.g005>

Therefore, we have access to the location details (via Foursquare API) and the time of check-ins (via Twitter API). Using the Twitter search API we found active users with high check-ins rate in Foursquare. By querying the API with “I am at”, the default template of Swarm app for check-ins, we selected the top 12000 users, and crawled their tweets in ten weeks during the year 2015. We pruned the data by selecting 1000 active users that were in the same country (Brazil). The average degree of the network is 6.4. The total number of check-ins is about 60000. The number of unique locations is about 10000 in 10 categories. Our data collection method complies with the terms of service of both Twitter and Foursquare. Moreover, the dataset is anonymized and does not reveal the identity of actual users.

We use the first eight weeks of the check-ins for train, and the remaining two weeks for test. The hyper-parameters of the temporal model are set to  $\tau = 24$  and  $\sigma = 1$ , by cross validation. We learn model parameters by the train data and use different temporal and spatial measures for the evaluations. We compare our proposed model with MH [17], where the intensity of user’s check-ins is modeled by a multivariate Hawkes process (the intensity depends on the user and her friends’ history); RNN [30] which use a recurrent neural network to learn a non-linear intensity function based on the users’ history of events; and baseline HP where the intensity is modeled by a Hawkes process that also depends on the user’s history. The spatial model is also compared with two baselines, MP and PL. In the MP method the most checked in locations, disregarding the time of check-ins, are more probable to be selected as the next check-in location. The PL model assumes periodicity in the location of check-ins, the locations that are more checked in previous periods are more probable to be visited in the current time.

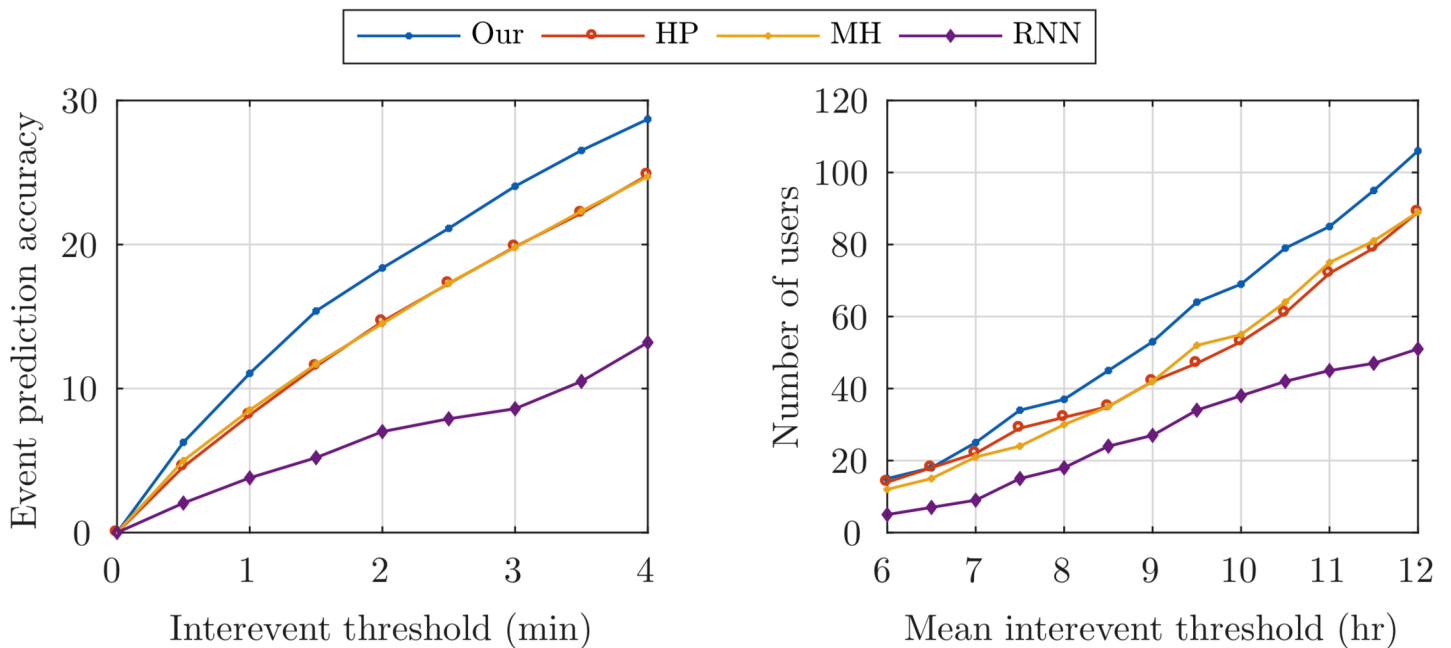
To reveal the motivation of the proposed method, we perform two empirical experiments on the real data. In summary, Fig 6 shows that: (i) most of the events are repeated after one, or more days (since there are peaks in the left graph at 1, 2, 3, . . .), which verifies the use of a periodic point process for modeling the time of users’ check-ins; (ii) about 80% of users are affected by their friend’s location of check-ins (the blue box) which justifies the use of the proposed mutually-exciting spatial model; (iii) only 10% of users explore new locations (the red box), which these users are modeled by the parameter  $\eta$  in Eq (13); (iv) as we more increase the size of the history time window, the less *Sociality* increases, which validates the use of the exponential decaying kernel in Eq (11) to reduce the effect of far past history.

To evaluate the prediction accuracy of the time of check-ins, we design two experiments. For each test event we estimate the time of the next event by different methods. The percent of check-ins which their times are closer than a threshold to the real time is plotted in the left



**Fig 6. Interevent time and sociality in real data.** The frequency of interevent times in the Food category of Foursquare dataset (left), and the Sociality box plot of users for different history window sizes (right).

<https://doi.org/10.1371/journal.pone.0197683.g006>

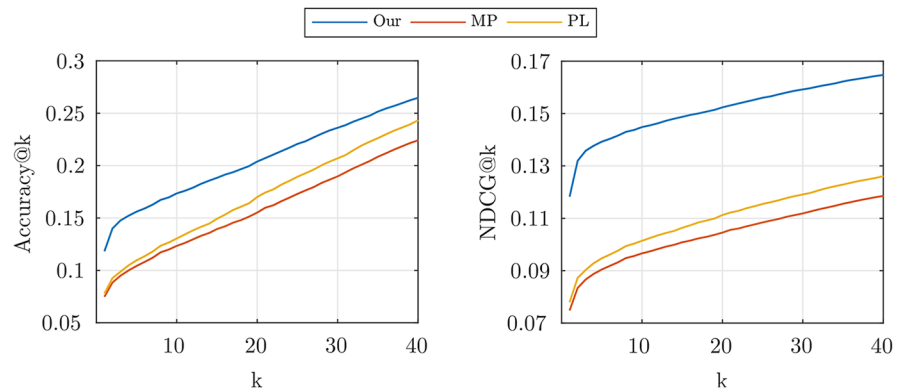


**Fig 7. Real data temporal measures.** The percent of check-ins which their times are closer than a threshold to the real time (left). The number of users which their average distance of predicted check-in times to the real times are less than a threshold (right).

<https://doi.org/10.1371/journal.pone.0197683.g007>

graph of Fig 7. Our method achieved up to 35% improvement for a one hour threshold, compared to other methods. In the right graph, the number of users where the average distance of their estimated events is less than a threshold is plotted. The proposed method performed up to 20% better than the competing methods. We did not plot this graph for the thresholds less than 6 hr, where all methods perform poorly. The poor performance of the RNN method is probably due to underfitting, since its objective function is nonconvex (in contrast to the other





**Fig 8. Real data spatial measures.** The accuracy (left) and NDCG (right) of location prediction, given the times of check-ins, at different values of  $k$ .

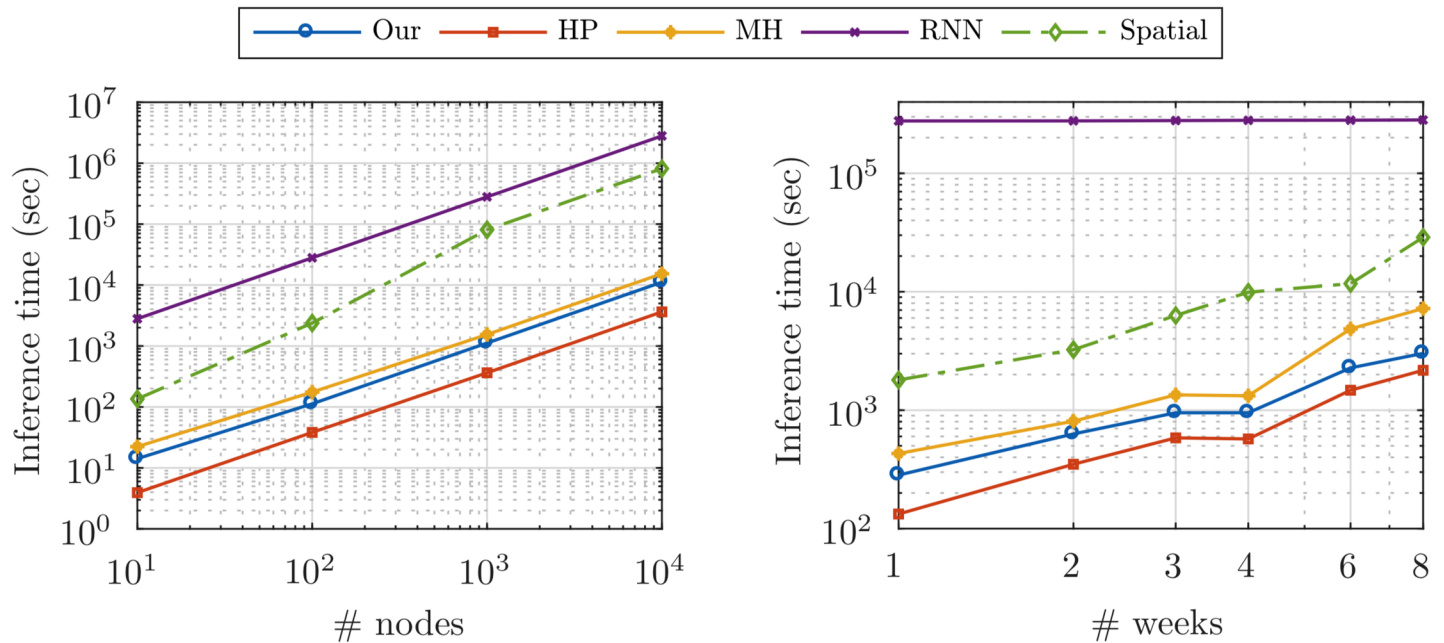
<https://doi.org/10.1371/journal.pone.0197683.g008>

methods, which are all convex), and the SGD method for the inference need much more training check-in data, which is rare in most of the real-world applications.

Now, given the time of check-ins, we evaluate the prediction accuracy of the location of check-ins. For each test event, each method assigns a probability to each location, forming a vector and selects the most probable location.  $Accuracy@k$  is the percent of events that the true location is among the first  $k$  high probable locations, and  $NDCG@k$  is

$\frac{1}{N} \sum_{i=1}^N \mathbb{I}(1 + r(e_i) < k) / \log_2(r(e_i))$ , where  $r(e_i)$  is the (one-based) rank of the real location of  $i$ th check-in in the location probability vector. These measures are plotted in Fig 8. For  $k = 1$  the accuracy increase from  $\sim 7\%$  in other methods to  $\sim 11\%$  in our method—about 43% improvement. It should be noted that there are about 10,000 locations and the random guess has extremely low accuracy. For larger values of  $k$  the measure is less reliable, since all method would have the same accuracy. Our method reaches to 24% accuracy, and about 8% improvement at  $k = 40$ . But in the  $NDCG$  which dose not have the mentioned undesirable effect (since the low-rank events are more significant) we see our method consistently outperform the others—about 30 to 50% improvement for the different values of  $k$ .

Finally, we performed the scalability analysis for different methods as depicted in Fig 9. In the right graph we compared the inference time for different sizes of event history, in the real dataset. Our method achieved the second best performance. For better comparison, the time complexity of all models, except RNN, are measured on a single core machine, although our method and HP can be executed in parallel and consequently the CPU time would be divided by the total number of cores. The time of RNN method is multiple orders of magnitude slower than the others, although we executed it on a 10-core machine, since the SGD methods need much more iterations to converge. Moreover, if we fit a line to these log-log curves, the slopes of Our, HP, MH, RNN, and Spatial curves would be 1.1, 1.3, 1.4, 0.01 and 1.2, respectively. This, validates the linear time complexity of our model with respect to the size of history  $h$ . In the left graph we compared the inference time in the synthetic data with different network sizes. Again, our method is the best performer after HP. Here, the slopes are 0.96, 0.98, 0.91, 0.99 and 1.2 for Our, HP, MH, RNN, and Spatial methods, respectively. These results validate the linear time complexity of our model with respect to the size of network  $N$ .



**Fig 9. Scalability comparison.** The time complexity of different temporal models and our spatial model (the other baseline spatial models also have approximately the same time complexity, so only one of them is depicted), for different network sizes (*left*), and for different sizes of events history (*right*).

<https://doi.org/10.1371/journal.pone.0197683.g009>

### Conclusion

To model the check-ins of users in location-based social networks, we proposed a doubly stochastic point process for the time of check-ins, which leverages the periodicity in users’ behavior, and a multinomial distribution for the location of check-ins, which leverages the mutually-exciting effect of friends on decision of users.

The synthetic experiments show the proposed inference algorithm can learn the model parameters with high accuracy and its performance increases consistently by the size of train data. Moreover, we study the effect of model parameters on the users’ check-ins, from which one can interpret the users’ behavior in LSBNs from their inferred parameters. The real experiments on the curated Foursquare check-ins dataset, show the proposed method outperform the other competing methods in the time and location prediction of users’ check-ins. Specifically, we achieved up to 35% in the time prediction and 43% in the location prediction accuracy. Furthermore, the empirical studies show the real data meets the assumptions of the proposed model that is, users are periodic in the time and mutually-exciting in the location of their checkins.

Our work also opens many interesting venues for future works. For example, we can consider the home location of the users in defining the probability of the location of their check-ins, by modifying the weight of locations in Eq (11). In addition, we can investigate the utilization of a non-parametric spatial model instead of the multinomial distribution. Finally, we can use the proposed model to control the check-in behavior of users by incentivization, or use it for point-of-interest recommendations.

## Author Contributions

**Conceptualization:** Hamid R. Rabiee.

**Formal analysis:** Ali Zarezade.

**Investigation:** Sina Jafarzadeh.

**Methodology:** Ali Zarezade.

**Software:** Ali Zarezade, Sina Jafarzadeh.

**Supervision:** Hamid R. Rabiee.

**Validation:** Sina Jafarzadeh.

**Writing – original draft:** Ali Zarezade.

**Writing – review & editing:** Hamid R. Rabiee.

## References

1. Zheng Y. Location-based social networks: Users. *Computing with Spatial Trajectories*, Zheng Yu and Zhou Xiaofang, Eds. Springer; 2011.
2. Zheng Yu Xing Xie WYM. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data(base) Engineering Bulletin*. 2010; 33(2):32–39.
3. Andris C. Integrating social network data into GISystems. *International Journal of Geographical Information Science*. 2016; 30(10):2009–2031.
4. Li Q, Wei W, Xiong N, Feng D, Ye X, Jiang Y. Social media research, human behavior, and sustainable society. *Sustainability*. 2017; 9(3):384. <https://doi.org/10.3390/su9030384>
5. Zheng Y. Trajectory Data Mining: An Overview. *ACM Transaction on Intelligent Systems and Technology*. 2015;.
6. Bao Jie Yu Zheng DWMFM. Recommendations in Location-based Social Networks: A Survey. *Geoinformatica*. 2014; 19:525–565. <https://doi.org/10.1007/s10707-014-0220-8>
7. Cho YS, Galstyan A, Brantingham PJ, Tita G. Latent self-exciting point process model for spatial-temporal networks. *Discrete and Continuous Dynamical Systems—Series B*. 2014; 19(5):1335–1354. <https://doi.org/10.3934/dcdsb.2014.19.1335>
8. Cho E, Myers SA, Leskovec J. Friendship and mobility: user movement in location-based social networks. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM; 2011. p. 1082–1090.
9. Williams MJ. *Periodic patterns in human mobility*. Cardiff University; 2013.
10. Zhenhui Li JHRKPN Bolin Ding. Mining periodic behaviors for moving objects. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010)*. ACM; 2010. p. 1099–1108.
11. Zarezade A, Khodadadi A, Farajtabar M, Rabiee HR, Zha H. Correlated Cascades: Compete or Cooperate. In: *Thirty-First AAAI Conference on Artificial Intelligence*; 2016.
12. Farajtabar M, Du N, Gomez-Rodriguez M, Valera I, Zha H, Song L. Shaping social activity by incentivizing users. In: *Advances in neural information processing systems*; 2014. p. 2474–2482.
13. Gomez-Rodriguez M, Song L, Daneshmand H, Schoelkopf B. Estimating diffusion networks: Recovery conditions, sample complexity & soft-thresholding algorithm. *Journal of Machine Learning Research*. 2015;.
14. Gomez-Rodriguez M, Balduzzi D, Schölkopf B. Uncovering the Temporal Dynamics of Diffusion Networks. In: *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*; 2011. p. 561–568.
15. Gomez-Rodriguez M, Leskovec J, Schölkopf B. Modeling Information Propagation with Survival Theory. In: *Proceedings of The 30th International Conference on Machine Learning (ICML'13)*; 2013. p. 666–674.
16. Iwata T, Shah A, Ghahramani Z. Discovering latent influence in online social activities via shared cascade poisson processes. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM; 2013. p. 266–274.

17. Yang SH, Zha H. Mixture of mutually exciting processes for viral diffusion. In: Proceedings of the 30th International Conference on Machine Learning (ICML'13); 2013. p. 1–9.
18. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM. Who, where, when and what: discover spatio-temporal topics for twitter users. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2013. p. 605–613.
19. Linderman SW, Adams RP. Discovering Latent Network Structure in Point Process Data. In: ICML; 2014. p. 1413–1421.
20. Gomez-Rodriguez M, Song L, Du N, Zha H, Schölkopf B. Influence Estimation and Maximization in Continuous-Time Diffusion Networks. *ACM Transactions on Information Systems (TOIS)*. 2016; 34(2):9. <https://doi.org/10.1145/2824253>
21. Hawkes AG. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*. 1971; 58(1):83–90. <https://doi.org/10.1093/biomet/58.1.83>
22. Valera I, Gomez-Rodriguez M. Modeling adoption and usage of competing products. In: Data Mining (ICDM), 2015 IEEE International Conference on. IEEE; 2015. p. 409–418.
23. He X, Rekatsinas T, Foulds J, Getoor L, Liu Y. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In: Proc. 32nd Intl. Conf. on Machine Learning; 2015.
24. Tran L, Farajtabar M, Song L, Zha H. NetCodec: Community Detection from Individual Activities. In: SIAM International Conference on Data Mining (SDM). SIAM; 2015.
25. Gui H, Sun Y, Han J, Brova G. Modeling topic diffusion in multi-relational bibliographic information networks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM; 2014. p. 649–658.
26. Hosseini SA, Khodadadi A, Arabzade S, Rabiee HR. HNP3: A Hierarchical Nonparametric Point Process for Modeling Content Diffusion over Social Media. In: Data Mining (ICDM), 2016 IEEE International Conference on. IEEE; 2016.
27. Zarezade A, Upadhyay U, Rabiee HR, Gomez-Rodriguez M. Redqueen: An online algorithm for smart broadcasting in social networks. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM; 2017. p. 51–60.
28. Zarezade A, De A, Rabiee H, Rodriguez MG. Cheshire: An Online Algorithm for Activity Maximization in Social Networks. In: 55th Annual Allerton Conference on Communication, Control, and Computing; 2017.
29. Du N, Dai H, Trivedi R, Upadhyay U, Gomez-Rodriguez M, Song L. Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM; 2016. p. 1555–1564.
30. Xiao S, Yan J, Yang X, Zha H, Chu SM. Modeling the Intensity Function of Point Process Via Recurrent Neural Networks. In: AAI; 2017. p. 1597–1603.
31. Xiao X, Zheng Y, Luo Q, Xie X. Inferring social ties between users with human location history. *Journal of Ambient Intelligence and Humanized Computing*. 2014; 5(1):3–19. <https://doi.org/10.1007/s12652-012-0117-z>
32. Liu B, Xiong H. Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness. In: Proceedings of the 2013 SIAM International Conference on Data Mining. SIAM; p. 396–404.
33. Scellato S, Noulas A, Mascolo C. Exploiting place features in link prediction on location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2011. p. 1046–1054.
34. Liu B, Fu Y, Yao Z, Xiong H. Learning geographical preferences for point-of-interest recommendation. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2013. p. 1043–1051.
35. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM. Time-aware point-of-interest recommendation. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM; 2013. p. 363–372.
36. R H Huayu Li YG, Zhu H. Point-of-interest recommendations: Learning potential check-ins from friends. In: Proceedings of the 22th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2016.
37. Tang LA, Zheng Y, Yuan J, Han J, Leung A, Hung CC, et al. On discovery of traveling companions from streaming trajectories. In: 2012 IEEE 28th International Conference on Data Engineering. IEEE; 2012. p. 186–197.
38. Zheng K, Zheng Y, Yuan NJ, Shang S, Zhou X. Online discovery of gathering patterns over trajectories. *IEEE Transactions on Knowledge and Data Engineering*. 2014; 26(8):1974–1988. <https://doi.org/10.1109/TKDE.2013.160>

39. Lichman M, Smyth P. Modeling human location data with mixtures of kernel densities. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2014. p. 35–44.
40. Lee JG, Han J, Li X. Trajectory outlier detection: A partition-and-detect framework. In: 2008 IEEE 24th International Conference on Data Engineering. IEEE; 2008. p. 140–149.
41. Liu S, Ni LM, Krishnan R. Fraud detection from taxis' driving behaviors. *IEEE Transactions on Vehicular Technology*. 2014; 63(1):464–472. <https://doi.org/10.1109/TVT.2013.2272792>
42. Yu Zheng LWXX Like Liu. Learning Transportation Mode from Raw GPS Data for Geographic Application on the Web. In: WWW; 2008.
43. Ogata Y. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association*. 1988; 83(401):9–27. <https://doi.org/10.1080/01621459.1988.10478560>
44. Barabasi AL. *Network Science*. Cambridge university press; 2015.
45. Daley DJ, Vere-Jones D. *An Introduction to the Theory of Point Processes—Vol. I*. Springer Ser. Statist., Springer, New York; 2002.
46. Aalen O, Borgan O, Gjessing H. *Survival and event history analysis: a process point of view*. Springer Science & Business Media; 2008.
47. Cox DR, Isham V. *Point processes*. vol. 12. CRC Press; 1980.
48. Kingman JFC. *Poisson processes*. Oxford university press; 1992.
49. Boyd S, Vandenberghe L. *Convex optimization*. Cambridge university press; 2004.
50. Leskovec J, Chakrabarti D, Kleinberg J, Faloutsos C, Ghahramani Z. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*. 2010; 11:985–1042.
51. Ogata Y. On Lewis' simulation method for point processes. *Information Theory, IEEE Transactions on*. 1981; 27(1):23–31. <https://doi.org/10.1109/TIT.1981.1056305>