Geoinformatica
https://doi.org/10.1007/s10707-020-00400-3

Learning evolving user's behaviors on location-based social networks

Ruizhi Wu[1] · Guangchun Luo[2] · Qi Jin[1,3] · Junming Shao[1] · Chang-Tien Lu[4]

Received: 23 June 2019 / Revised: 16 December 2019 / Accepted: 4 February 2020 /
Published online: 16 March 2020

Abstract
With the popularity of smart phones, users' activities on location-based social networks (LBSNs) evolve faster than traditional social networks. Existing models focus on modeling users' long-term preferences, leveraging social collaborative filtering to enhance prediction performance. However, the dynamic mobility mechanism of user's check-in behaviors on LBSNs is seldom considered. In this paper, we propose a new dynamic model that considers both geo-aware user preferences and the social interaction excitation arising from social connections to learn the dynamic mobility mechanism of user's behaviors on LBSNs. Geo-aware location features, such as semantic features, latent features and dynamic features, are utilized to characterize the location information and reveal the evolution of the geographical impact of location. These geo-aware location features enable us to exploit user's personal preferences. Meanwhile, we integrate a user's social connections and friends' preferences for modeling social interaction excitations. Finally, we jointly incorporate geo-aware user preference learning and social interaction excitation modeling to create a conditional intensity function for temporal point processes with which to explore the dynamic mobility mechanism of evolving user's check-in behaviors on LBSNs. Extensive experiments on several real-world check-in datasets confirm that our proposed algorithm performs better than existing state-of-the-art methods.

Keywords LBSNs · Dynamic model · Temporal point process

1 Introduction

Successful products in the mobile Internet era have a strong social element, with most social networks providing location-based services. For instance, Gowalla, Brightkite, and Foursquare allow users to share their experiences with friends when they visit a location [17]. Oceans of check-in data from smartphones and websites carry user's mobility preferences, time schedule, and social information [32, 44], hence the ability to accurately predict

✉ Guangchun Luo
gcluo.uestc@gmail.com

Extended author information available on the last page of the article.

the location and time of a user's next check-in activity could have many interesting applications. For instance, if an advertiser could successfully predict the location and time of a user's check-in activity they could provide them with more relevant and timely information and marketers could select the optimum location and time to present them with a personalized promotion or special offer. For government agencies, predicting users' future activity would facilitate traffic control and provide early warnings of potential public emergencies. In this paper, we aim to model evolving user behaviors to predict the location and time of a user's next check-in activity. Although a great deal of progress has been achieved in recent years, learning a user's check-in behavior on LBSNs is still a non-trivial task due to the following three potentially serious challenges.

(1) Geo-aware User Preference Learning. Previous studies have indicated that the motivations driving a user's check-in behavior falls into three main categories: 1) participating in an event, such as at a casino, or in a hospital; 2) the user's inherent preferences; and 3) their wish to follow a trend [2, 5, 17, 22, 33, 45]. However, existing models of user behavior focus only on the user's inherent preferences, totally ignoring factors 1) and 3), both of which are related to geographical features of the location itself. Another drawback is that existing methods typically use longitude and latitude directly as geographical location features [3, 49], which creates two main problems: 1) the lack of appropriate semantics leads to faulty interpretations as we simply cannot understand why people are visiting this location; and 2) the longitude and latitude values alone provide no useful information regarding the geographic evolution of a location, such as the fashion trend of the location, its attraction for new users, the average time intervals typically linked to user check-in behavior at the location, and so on. So, how can we best exploit the full potential of having geographical information for a particular location?

(2) Dynamic Excitation of Social Interaction. Nowadays, LBSNs provide a convenient way to share experiences among people via geo-location tags. Their interactions with social networks will affect users' check-in behavior as users share their life experience with geo-location tags. Early studies have generally characterized social influence as a fixed factor that can be utilized to improve a model's performance and enhance its robustness [36]. However, a fixed social factor cannot reflect real world variations in social influences. We refer to these variations in social influences as dynamic excitations of social interaction. The analysis of dynamic changes in social interactions is more difficult than leveraging social influences as a fixed factor. Thus, the dynamic influence of social interactions over time has not yet been thoroughly investigated. At the same time, most existing approaches only consider direct social interactions, with little attention being paid to the high-order social interactions in social networks.

(3) Dynamic Mobility Mechanism. As people's daily mobility activities have become increasingly frequent, so have their check-in behaviors changed over time. For example, according to statistics presented in the Brightkite data set, during the competition season, the number of check-in records for gymnasts and their supporters in or near to their gymnasiums exceeds those during the non-competition season by more than 57.8% and is 27.3% higher than their yearly average. The number of check-in activity drops back to normal after the competition season ends. This is also visible in the analysis of the check-in data sets for California and Tokyo shown in Fig. 1, which depicts the normalized value of average visiting frequency for all locations over a six month period. It is clear that changes in the visiting frequency happen in just a few days or weeks. All these phenomena are manifestations of the dynamic evolution of

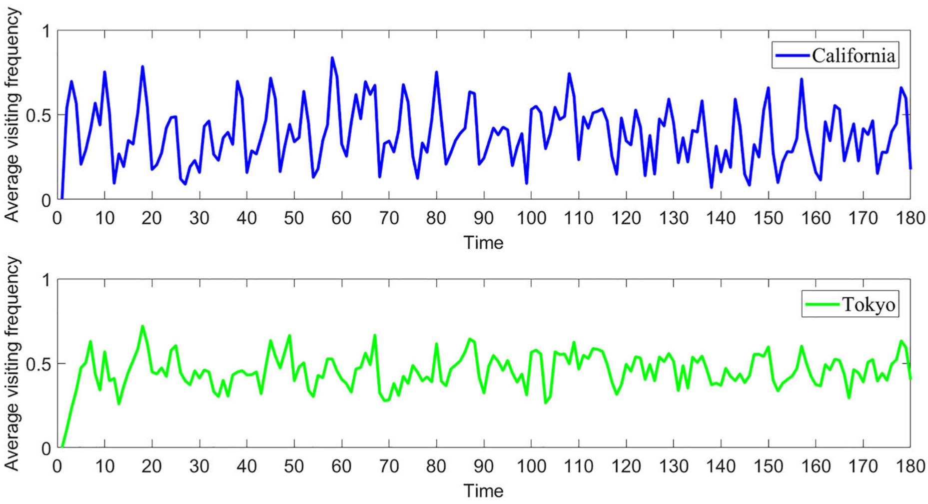


Fig. 1 Normalized values of the average visiting frequency of all locations for the 6 months in California (from January 1 to June 30, 2009) and Tokyo (from July 1 to December 31, 2012) check-in data sets

user behaviors on LBSNs. These dynamic evolutions are not only caused by changes in user preferences but are also due to the impact of social interactions and changes in the frequency of check-in activity. Studying dynamic evolution is thus helpful for modeling evolving user behaviors over a long period of time. The key objective when studying the above dynamic evolution is the dynamic mobility mechanisms driving the user behavior, including the context dependency and dynamic change modeling.

Existing methods either do not address these challenges at all or fail to address them effectively [2, 35, 51]. For instance, the idea behind existing preference learning models is to represent users' inherent preferences in a lower dimensional space, where the assumption is that preferences either do not change or change only slowly, typically utilizing models based on matrix or tensor factorization. As mentioned in Challenge 1, fashion trends change rapidly over time, and existing static preferences learning models find it impossible to capture such fast-moving changes. One limit of existing static preferences learning models, such as matrix or tensor factorization method, is that it cannot be incrementally updated if the underlying data distribution has dramatic changes. Also, existing models are unable to integrate dynamic social influences because they ignore the way that social influences change along with crowd's preferences, which change constantly instead of remaining stable. For example, RCH [34], HPY [12] and CEPR [19] all exploit social influences as a static weight or fixed vector. Existing models are thus unable to capture the changes in user preferences and social interactions, both of which affect user check-in behaviors. Furthermore, existing models fail to explore the dynamic mobility mechanisms governing the evolution of user check-in behaviors as they ignore their context dependency and occurrence rate.

From a technological point of view, existing methods are not suitable for modeling evolving user behaviors. For instance, classic matrix factorization or tensor factorization methods cannot model changes in user preferences and the temporal dependency of user behaviors. Although Markov models, which are a generalized method widely used to model context dependency, fail to capture long-term dependency when the number of states is large. This is because the overall state-space grows exponentially with the number of time steps considered. Recently, LSTM has shown considerable promise for learning context dependency; existing models based on LSTM aim to integrate a user's preferences and social influences to recommend locations to him or her. However, models based on deep learning normally lack the type of intuitive interpretations needed to explain the underlying reasons for an individual user's check-in behavior and the dynamic changes in it. Also, deep learning models need a huge training data and their results are often unsatisfactory when forced to deal with data sparsity.

In this paper, we propose a novel framework DGPS for studying evolving user behaviors on LBSNs, in which a **d**ynamic model jointly performs **g**eo-aware personal **p**reference learning and **s**ocial interaction excitation modeling. In this new framework, users' check-in behaviors are considered in chronological order as a temporal point process. Temporal point process with the intensity function provides a common mathematical framework for modeling time-dependent dynamic evolution processes [8], such as earthquakes and aftershock data. Each check-in activity performed by a user can be represented as a point in temporal-spatial space, as illustrated in Fig. 2, which allows us to adopt a temporal point process to capture the context dependency and dynamic changes in these check-in activities. To model evolving user behaviors, the conditional intensity function is leveraged utilizing an inhomogeneity temporal point process to model instantaneous rate of the user's check-in activities, for example the second of the user's check-in activities at time $t4$ (enjoying drinking at a bar). Here, the user's personalized preference learning and social interaction excitation are incorporated into the conditional intensity function, in this case his drinking preferences (red broken dot lines) and friends' influences via social networks (green dashed lines). Knowing the motivation for a user's behavior, we can then build a unique geo-aware location feature engineered to characterize the location as a vector and thus exploit that user's personalized preferences. In this case, the geo-aware features of the location consist of three components matched to corresponding motivation categories: 1) Semantic features, 2) Latent features, and 3) Dynamic features, as illustrated in Fig. 3. This allows us to introduce a preference coefficient to describe the contribution of each of the three categories of geo-aware location features based on the user's preferences. To study the impact of social factors on the user's

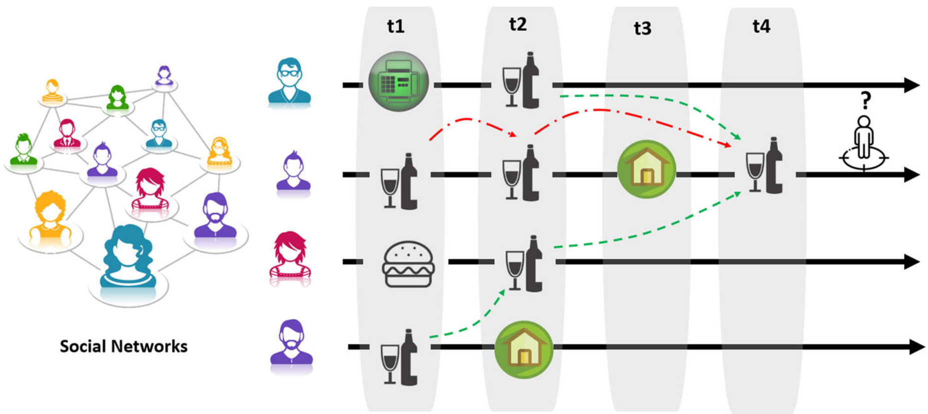


Fig. 2 The second user's check-in behavior arises from personal preferences (red broken line) and exterior interaction excitation (green dashed line). Here, the exterior interaction excitation comes from the user's social connections. DGPS aims to forecast the location and time of the user's future check-in activities

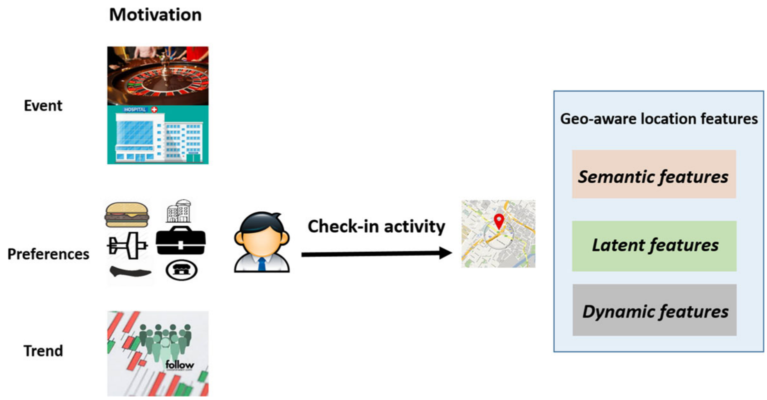


Fig. 3 Inspired by the motivation of a user's behavior, we build a unique geo-aware location feature that consists of semantic features, latent features, and dynamic features

behaviors, we can integrate social structure information and friends' behavior preferences to model the user's social interaction excitations via a triggering term. The high order social interaction is preserved via a propagating method (see the bottom user effect on the second user in Fig.2) even though there is no direct link. Finally, our proposed DGPS model adopts the ADMM optimization framework to learn the parameters of the proposed model in a supervised way. Our proposed approach thus contributes to activity prediction by taking into account the evolving user's check-in behavior, covering both location and corresponding time, i.e. predictive behavior modeling. The key idea here is to model the instantaneous rate of the user's check-in activities by tracking the conditional intensity function for the temporal point process and parameterizing the conditional intensity function for the user's preference learning and dynamic social excitation.

The main contributions of this paper are summarized as follows.

- **Extracting Geo-aware location features for learning user preferences.** We propose a geo-aware location feature engineering based on Google map service, non-negative matrix factorization and several specific extraction methods. Unlike traditional geographical features, geo-aware location features involve an understandable semantic expression, the representation of latent factors, the user distribution entropy of location, and the ratios of new users, and so on. By extracting geo-aware location features, DGPS provides deep insights into user preferences.
- **Modeling evolving user check-in behaviors on LBSNs.** To address the challenge of fast evolution of user behaviors, we propose DGPS for modeling evolving user preferences and social interaction excitations based on a temporal point process. DGPS integrates user' personal preferences and social interaction excitations to create a conditional intensity function for the temporal point process, which can then be applied to explore the dynamic mobility mechanism of user check-in behaviors on LBSNs.
- **High Performance.** By extracting geo-aware location features and leveraging a temporal point process to model user check-in behaviors on LBSNs, DGPS supports high prediction performance for user check-in behaviors. We have conducted experiments on several real-world data sets to demonstrate the performance of our proposed new DGPS model. The experimental results show that our proposed model delivers a markedly better performance than many of the existing state-of-art methods.

The remainder of this paper is organized as follows: The following section briefly surveys related work. Section 3 gives the intuition and whole framework of DGPS model. Section 4 presents the process of geo-aware location features extraction. Section 5 presents our proposed dynamic model and learning process in details. Section 6 contains an extensive experimental evaluation. Finally we provide a brief discussion and conclude the paper in Section 7.

2 Related works

In recent years, the studies on location-based social networks achieve impressive achievement. We review highly related works. In addition, we introduce some related works about temporal point process.

2.1 LBSNs model

During the past decade, many approaches have been proposed for modeling user's check-in behaviours on LBSNs, we discuss existing models from different aspects.

Mobility pattern mining. Early studies focused on regular pattern of user's check-in behavior, such mobility pattern and spatial-temporal pattern. For example, Monreale et al. built a decision tree for scoring mobility pattern [28], and Gambs et al. proposed Mobility Markov Chain (MMC) for modeling the transfer probability between different locations in check-in historical data [11]. The performance of pattern-based methods depended on extracting meaningful patterns, which were a overly exacting task. Pattern-based methods also seem have poor robustness and lack diversity in check-in locations.

User preference learning. Inspired by recommendation systems, some researchers realized the importance of user preferences and geographical influence on modeling user behavior. To analyse the preferences of user's check-in behavior, the matrix factorization models have been widely extended to user-location factorization models by incorporating the geographical influences. Two of the representative was GeoMF [21] and RCH [34]. Lian et al. proposed a weighted matrix factorization framework (GeoMF), which leveraged spatial clustering phenomenon derived from geographical influence for addressing the challenge of data sparsity. Wang et al. proposed regularity and conformity heterogeneous (RCH) algorithm, which jointly performed user's personal preferences and exterior popular preferences into a uniform matrix factorization framework. In which, RCH introduced heterogeneous spatial data for modeling exterior popular preferences, such as bus data, taxi data and check-in data.

Social influence. Many recent studies showed the combination of social connections and geographical influence was helpful to improve performance of prediction or recommendation model of user's check-in activities [18, 22, 50]. For example, Gao et al. proposed Hierarchical PitmanYor (HPY) process model to assesses the role of social networks in user's check-in behavior which aimed to well explain user's mobility activities [12, 13]. Yuan et al. proposed a W^4 (Who + Where + When + What) jointly probabilistic graph model which took into consideration of some factors, such user's profile information, check-in location, check-in time and twitter content, to predict user's mobility activity [48]. Lian et al. incorporated geo-location features as geographical influences into the proposed collaborative exploration and periodically returning (CEPR) model [19]. Jia et al.

analysed the influence of social connections theoretically, and proposed a Bayes networks for maximizing social influences and benefiting the performance of model [15]. Besides, Xu et al. studied the dynamic influence in social group and model user's willingness on event participation, which aims to predict user's social events participation [42, 43].

Time effect. The influence of temporal information of user's check-in event were another hot topics. The time effect modeling of user's behavior mainly leverage temporal periodic pattern, sequences pattern and tensor factorization. Xu et al. proposed a time series model for mining user multiple periodic behaviors, and incorporated periodic pattern into collective filtering model to recommend location [40]. To model time effect in intuitive way, the matrix factorization models have been widely extended to time effect tensor factorization models by incorporating the temporal information, such as multi-dimensional collaborative recommendations model [3], the regularized content-aware tensor factorization (RCTF) [20]. Liu et al. proposed WWO to capture the user's sequential preferences with temporal interval assessment for recommending new locations to the user [25].

Deep learning models. Recently, deep learning techniques have great achievements in area of NLP, CV and sequence analysis. Some works delivered deep learning model into research area of LBSNs model for prediction and recommendation tasks. Liu et al. proposed spatial temporal recurrent neural networks (ST-RNN) which modeled time-specific transition matrices and distance-specific transition matrices respectively for studying correlation between geographical influences and temporal effect [23]. Yin et al. adopted a deep learning model to learn the location representation from heterogeneous feature and a user's preferences for POIs recommendation in home-town or out-of-town scenarios [46, 47].To overcome data sparsity problems, they formulated social regularization and region-depend smoothing objective functions.

As described above, although there were many studies to learn user's behavior model by exploring geographical influences, social connections and time effect, these works did not address the aforementioned challenges (e.g., evolution of geographical influences, dynamic excitation of social interaction and geo-aware user preferences learning) arising from fast evolving user's check-in activities on LBSNs. Most of the above model focused on geographical influences and social connections and ignore the equally crucial of time factors. It is a fact that time effect are now viewed not a simply factors with a role in affecting user's check-in behavior, but as a leading actor that addresses the problem of evolving user's behavior on LBSNs. To combat the aforementioned challenges in Section 1, we propose DGPS that jointly perform geo-aware user's personal preferences learning and social interaction excitation modeling into a temporal point process for studying dynamic mobility mechanism of user's behavior on LBSNs.

2.2 Temporal point process

Temporal point processes are mathematical tools in statistics and probability, and they have been widely used for modeling many different phenomena in many domains such as seismology, computational finance and sociology [6, 24, 30, 37, 38]. In [8, 39], Du et al. and Xiao et al. summarized some typical temporal point processes and corresponding parameterized intensity function. Recently, the temporal point process and its variants have been used to model the human activities. For example, Costa et al. used a temporal point process to model inter-arrival times (IATs), which aimed to perform prediction tasks on Reddit and Twitter [10]. Nan Du et al. incorporated a low-rank matrix factorization technique into a self-exciting point process for performing recommendation tasks [9]. Recurrent neutral networks have unique internal state (memory) to process sequences of inputs, and make impressive achievement on sequences analysis. Nan Du et al. proposed a recurrent marked temporal point process (RMTPP), in which RMTPP used RNN automatically learn the representation of sequence data given historical data [8].

Existing models focus on the specific tasks and domains, and not well model the evolving user behaviors on LBSNs. DGPS aims to explore user's check-in activities by considering user preferences and social interactions with a temporal point process. Different from existing models, DGPS adopts an intuitive way to learn user's personal preferences and social interaction excitations derived from social connections. In the following, we will first give the intuition and overview of DGPS.

3 Intuition and overview

To learn evolving user behaviors on LBSNs, an intuitive model jointly performs geo-aware personal preference learning and social interaction excitation modeling, and meanwhile, the dynamic mobility mechanism of user's check-in activities are captured by a dynamic model based on the temporal point process. The model consists of two closely related components, see Fig. 4. To mine geo-aware user's personal preferences, we build a geo-aware location feature engineering, which aims to access a representation of location and reveal evolution of geographical influences of location. In which, geo-aware location features includes three components: semantic features, latent features, and dynamic features. We extract geo-aware features for each location of user's check-in activities in chronological order, and notate $\{(f_{l1}, t_1), (f_{l2}, t_2), ...(f_{lT}, t_T)\}$, see the orange module in Fig. 4. On the basic of obtaining geo-aware location features, we leverage a parameter to learn user's

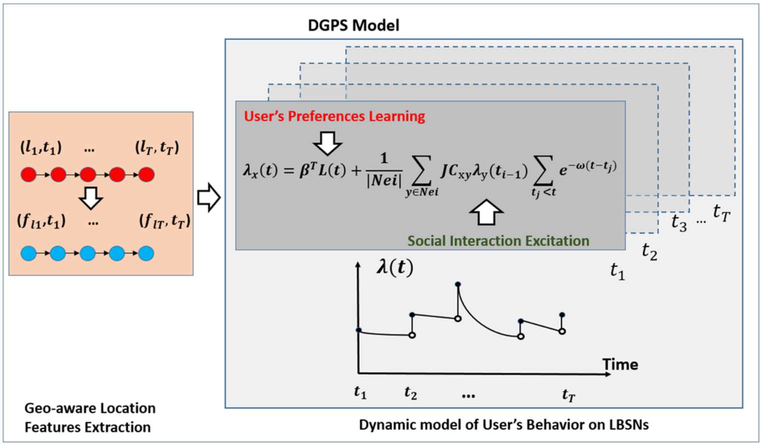


Fig. 4 The framework of our proposed method: DGPS. To learn user behaviors on LBSNs, DGPS first extracts geo-aware location features for every location $\{(f_{l1}, t_1), ...(f_{lT}, t_T)\}$ from the check-in activities $\{(l_1, t_1)...(l_T, t_T)\}$ in chronological order. Relying on the geo-aware location features, we further mine the user's geographical preferences. Finally, we incorporate user preference learning and social interaction excitations into a conditional intensity function for exploring dynamic mobility mechanism of user's check-in activities on LBSNs

geo-aware preferences, in which the user's preferences consider both inherent influence of location (e.g., semantic features and latent features) and evolution influences (dynamic features). To model social interaction excitation, we firstly measure users' similarity on social networks relying on Jaccard similarity coefficient, and we leverage the previous intensity of friends' check-in activity as friends' preferences. Another key point of social interaction modeling is to explore time effect of social influences. Therefore, by considering evolution of social influences, we integrate social similarity and friends' preferences as social interactions and leverage a time kernel function as a triggering term to model social interaction excitation. Finally, we leverage a conditional intensity function of temporal point process to model the dynamic evolution of user's check-in activities, illustrated as the dark gray module in Fig. 4. In which, the conditional intensity function jointly performs geo-aware user preference learning and social interaction excitation modeling. To learn the parameters, we use ADMM framework and maximize likelihood estimation method to solve our optimization problems. Before we start the description of our model, the main notations used in this paper is given in Table 1.

4 Geo-aware location features extraction

Extracting geo-aware location features aims to represent location as a vector and reveal the evolution of geographical influence of a given location. For example, see Fig. 5, the location marked with an asterisk in left side is a location in Washington DC. The geo-aware features of the location marked with an asterisk consist of three components (the right side): semantic features (orange module), latent features (green module), and dynamic features (grey module). To this end, this section presents the details of extraction procedure of geo-aware location features.

4.1 Semantic features

The category of the check-in location carries semantic information of user's check-in activity. To explore the semantic content of user's check-in behaviors, we query category labels

Table 1 Main Notations

Notations	Description
$\lambda^*(t)$	Conditional intensity function.
$L(t)$	Geo-aware location features.
β	The preference coefficient of location features.
$\mathbf{X}$	User-location frequency matrix.
$\mathbf{U}$	User preference matrix.
$\mathbf{V}$	Location latent features matrix.
$\mathbf{W}$	Weight matrix.
$\Gamma(\cdot)$	The set of user's neighbour nodes in social network.
ω	Time excitation kernel factors.
k	The number of latent factors.
t	Time slice, where the length is T.
i, j	Index label.

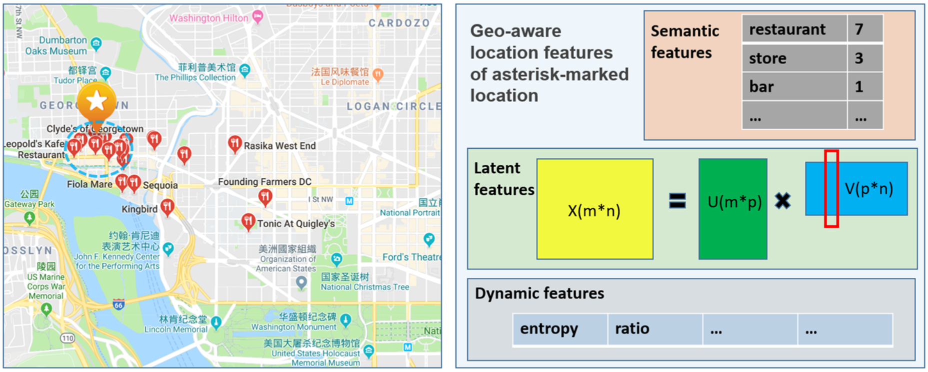


Fig. 5 Illustration of the geo-aware location features of a given location. The geo-aware location features of the location marked with an asterisk include semantic features, latent features and dynamic features. Geo-aware location extraction processes rely on Google map API, non-negative matrix factorization, and several specific extraction methods

of neighboring places around location by Google map API, which represent the semantic features of the location. Since the same category or function of locations always have more closer geographical distance in urban planing. It is reasonable that using category labels of neighboring places represent semantic information of a given location. For example, the value of longitude and latitude of Virginia Tech North Virginia Center (NVC) is (38.897378, −77.189441). The neighboring place query results have 9 labels within 150 meters: four of them are university, and three of them are school, the rest are store. And this result is consistent with that NVC is a research center of university. The location category list of Google map API shows in Table 2. Most locations can receive acceptable results within a radius of 150 meters, while the rest locations need to extend query radius to 500 or 1000 meters. Finally the average radius is 250 meters. According to the query result, we compute the number of occurrences of corresponding categories of location to formulate a fixed-length vector. To highlight the main semantic characteristic of location, we utilize softmax function on the vector, as in Eq. 1, where c_i is the number of occurrences of the i-th location category, s_i is corresponding item in semantic vector. The semantic vector s consists of all the s_i in the location categories list.

$$s_i = \frac{e^{c_i}}{\sum_j e^{c_j}} \tag{1}$$

4.2 Latent features

Matrix factorization gains impressive achievement as an example for mining latent factors of user preferences. In which, non-negative matrix factorization (NMF) fits to process a low-rank approximation of non-negative matrix. To explore user preferences, we leverage NMF to learn latent features by decomposing the user-location frequency matrix. Specifically, we construct a user-location frequency matrix, and decompose it into the product of two lower dimensional matrices. We use Eq. 2 to calculate the user-location frequency matrix.

$$x_{ij} = \frac{c_j}{\sum_L c} \tag{2}$$

Table 2 Location category label summaries

Location category list					
accounting	airport	amusement park	aquarium	art gallery	atm
bakery	bank	bar	beauty salon	bicycle store	book store
bowling alley	bus station	cafe	campground	car dealer	car rental
car repair	car wash	casino	cemetery	church	city hall
clothing store	convenience store	courthouse	dentist	department store	doctor
electrician	electronics store	embassy	fire station	florist	funeral home
furniture store	gas station	gym	hair care	hardware store	hindu temple
home goods store	hospital	insurance agency	jewelry store	laundry	lawyer
library	liquor store	local government office	locksmith	lodging	meal delivery
meal takeaway	mosque	movie rental	movie theater	moving company	museum
night club	painter	park	parking	pet store	pharmacy
physiotherapist	plumber	police	post office	real estate agency	restaurant
roofing contractor	rv park	school	shoe store	shopping mall	spa
stadium	storage	store	subway station	supermarket	synagogue
taxi stand	train station	transit station	travel agency	veterinary care	university
zoo					

Where c_j is the number of check-in occurrences of the j-th location, and $\sum_L c$ represents the whole number of check-in occurrences of all locations of i-th user. x_{ij} is the frequency of i-th user visits j-th location. $X \in \mathbb{R}^{m \times n}$ is user-location frequency matrix, where m is the number of users, and n is the number of locations. Considering the data sparsity problem, a weight matrix is introduced to overcome it, as in Eq. 3. Here w_{ij} is a binary weights depend on the value of x_{ij} is observed or unknown.

$$w_{ij} = \begin{cases} 1, & \text{if } x_{ij} \text{ is observed} \\ 0, & \text{if } x_{ij} \text{ is not observed.} \end{cases} \tag{3}$$

To access the representation of latent features of a location, we formulate an objective function as Eq. 4, and decompose the frequency matrix X into the product of two lower dimensional non-negative matrix U and V.

$$\begin{aligned} \min \quad & \|W \odot (X - UV)\|^2 \\ \text{s.t.} \quad & U \geq 0 \quad V \geq 0 \end{aligned} \tag{4}$$

where, $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{k \times n}$ are user preference matrix and latent feature matrix for location, respectively. k is the number of latent factors of non-negative matrix factorization.

Lee et al. proposed a multiplicative updates for weight non-negative matrix factorization with Eqs. 5 and 6 [16].

$$U \leftarrow U \odot \frac{[W \odot X]V^T}{[W \odot UV]V^T} \tag{5}$$

$$V \leftarrow V \odot \frac{U^T[W \odot X]}{U^T[W \odot UV]} \tag{6}$$

The matrix of V is the location representation in form of k latent components. Section 6 further shows the sensitivity of k against to model performance.

4.3 Dynamic features

Semantic features are inherent characteristic of location, and the latent features are the global representation of latent factors and can not change quickly. To study the evolution of geographical influence of location, we extract eight types of dynamic features. Firstly, we divide time lines into T intervals. From Fig. 1, we observe the time intervals of successive peaks of the average visiting frequency of locations are around 7 days. Thus, the time interval is set as one week (7 days) in this study. We chronologically order user's check-in activities and denote them as $\{(l_1, t_1), ...(l_T, t_T)\}$. In the following, we will extract features in each time unit for every location, and the superscript t represents time interval, and the subscript l represents the location.

a) **The distinct number of users for each location** $|C_l^t(\text{user})| \in \mathbb{N}$. It was also used in Song et al. [33]. The distinct number of users for each location on each time unit shows the geographical attraction of the location. With the increase number of distinct users, the location has more active geographical attraction, and vice versa. This quantity reflects the current status of attraction for a location.
b) **The number of check-ins** $|C_l^t(\text{check-in})| \in \mathbb{N}$. This quantity directly reflects the "heat" of a location and is a common feature used in [49].
c) **The ratio of check-ins on a location** $r_l^t \in [0, 1)$, which is the ratio of the number of check-ins at a location l to the total number of check-ins, given time t.
d) **User's entropy at the location** $H_l^t(user)$, which is calculated based on the user's check-in frequency at the location, given time t. Specifically, if the check-in frequency of i-th user is $p(i)$, the user's entropy can be calculated with Eq. 7. This entropy indicates the uncertainty of user distribution at this location.

$$H_l^t(user) = -\sum_i p(i) \log p(i) \tag{7}$$

e) **The new user ratio**, $r_l^t(\text{new user}) = \frac{c_l^t(\text{new user})}{c_l^t(\text{distinct user})}$, which is the ratio of number of new users to total number of distinct users, given the time t and the location l. Here, we can make a reasonable assumption that if the ratio increases over time, the geographical influence of the location will be enlarged at once, and vice versa.
f) **The ratio of the number of users in consecutive time**, $r_l(t_i|t_{i-1}) = \frac{c(user)^{t_i-1}}{c(user)^{t_i}}$, which is the ratio of number of distinct users at the previous time to number of distinct users at the current time, given a location l.
g) **The ratio of successive users** $r_l^t(user_s) = \frac{c(user_s)}{c(\text{distinct user})}$, which is the ratio of number of successive users to the total number of distinct usesr. The ratio reflects the long-term influences of location to loyal user.

h) **The average difference of time of successive check-in behaviors** δt, which is calculated by Eq. 8. In which, q is the number of users that have successive check-in behaviors.

$$\delta t = \frac{1}{q} \sum_{q} |t_i - t_{i-1}| \tag{8}$$

Above features reflect dynamic evolution of geographical influences of a location in term of the number of check-in activities, time and user's entropy, etc. So far, we jointly put semantic features, latent features, and dynamic feature in sequences for constructing geo-aware location features. Finally, we give the pseudocode of geo-aware location feature engineering in Algorithm 1. With these features at hand, we incorporate a learning model to explore geo-aware user's personal preferences in next step.

Algorithm 1 Location features engineering extracting algorithm.

Input:
Input Check-in activities datasets on LBSNs;
Output:
Output Check-in activities with geo-aware location features $\{(f_{l1}, t_1), (f_{l2}, t_2), ...(f_{lT}, t_T)\}$;
divide check-in dataset into T equal time intervals according to check-in timestamps, as $\{(l_1, t_1), (l_2, t_2), ...(l_T, t_T)\}$;
build location categories dictionaries.
for each location in Check-in activities datasets **do**
    search nearby place categories within average 250 meters radius;
    construct location categories vector;
    use Eq. 1 to normalize location categories vector for accessing semantic features s;
end for
use Eq. 2 to build user-location frequency matrix X and weight matrix W;
minimize Eq. 4;
while not converges **do**
    update U by Eq. 5;
    update V by Eq. 6;
end while
for each location in Check-in activities datasets **do**
    extract geo-location dynamic features using item (4.3) to item (4.3) in Section 4.3;
    construct geo-location dynamic features d;
end for
for each location in Check-in activities datasets **do**
    index latent features l in V;
    generalize the location dynamic features $f_l = [s, l, d]$;
end for
return Check-in activities with location features $\{(f_{l1}, t_1), (f_{l2}, t_2), ...(f_{lT}, t_T)\}$;

5 DGPS Model

In this section, we present DGPS. Firstly, we introduce the preliminary of conditional intensity function of temporal point process in following section. Sections 5.2, 5.3, and

5.4 present user's personal preference learning, social interaction excitation modeling, and dynamic modeling of user's behavior, respectively. Section 5.5 shows the learning and forecasting algorithm of DGPS. Lastly, we discusses the time complexity of DGPS in Section 5.6.

5.1 Preliminary

Temporal point process can be characterized by a conditional intensity function $\lambda^*(t)$, where the conditional intensity function reflects the expected instantaneous rate of user's activity under historical data, that is intensity of user's activity [1]. In general, the definition is given in Eq. 9.

$$\lambda^*(t) = \lim_{dt \to 0} \frac{E((N(t+dt) - N(t))|H(t))}{dt} \tag{9}$$

Where $\lambda^*(t)$ is the conditional intensity for the occurrence of a new event given the historical data $H(t)$, within a time window $[t + dt)$. $N(t)$ is the number of event, E is expected number of occurrence of new event. Transferring dt to the left side, as Eq. 10.

$$\lambda^*(t)dt = \mathbf{P}\{\text{a new check-in event within } [t, t+dt)|H(t)\} \tag{10}$$

$\lambda^*(t)dt$ indicates the occurrences probability of a new check-in event within time $[t, t+dt)$. * notation indicates intensity function relies on historical data.

The above formulation shows that the temporal point process is an effective way to model the occurrences of discrete events such as check-in data. This allows us to embed key factors known to affect a user's check-in behavior into a conditional intensity function, making it possible to model the mechanism generating that user's behavior. We can also now separate the key factor into two parts for learning user's preferences and exploring the mutual influences of social connections. Furthermore, we can leverage the temporal point process to capture temporal dependency of context of a user's check-in activity and model dynamic changes in their check-in behavior over a long period of time. This analysis of the context dependency and dynamic changes in a user's check-in behavior is helpful for efforts to accurately predict the location and time of a user's next check-in activity. Under the proposed framework, the reason why a user visits a location is clear and straightforward, and we can intuitively explore the dynamic mobility mechanism of their check-in behavior based on this knowledge of the relevant dynamic changes and context dependency. Therefore, to study the dynamic mobility mechanism of user's behaviors on LBSNs, DGPS utilizes conditional intensity function of temporal point process for modeling instantaneous rate of expected occurrences of user's check-in activities on LBSNs. The conditional intensity function, jointly encodes user's personal preference and social interaction excitations. In the following, we will introduce how to incorporate user's personal preference learning and social interaction excitation modeling into a conditional intensity function $\lambda^*(t)$.

5.2 User's personal preference learning

Many studies demonstrate that there is a strong correlation between the user's preferences and geographical influence of locations [2, 21, 46, 50]. Relying on geo-aware location feature engineering, we can model user's personal preferences with the product of the geo-aware location features and a preference parameter, as in Eq. 11. Here, $L(t)$ is the geo-aware location features on each time unit, and β is preference coefficient.

$$\lambda^*(t) = \beta^T * L(t) \tag{11}$$

Equation 11 reflects that user's check-in behaviors derived from user preferences. What is more, the instantaneous rate of user's check-in behaviors varies with geo-aware location features over time. We adopt a parameter β to learn the correlation between user preferences and geo-aware location features. Although the correlation may have negative value, it lacks reasonable explanation in real-world scenarios. Therefore, we set β to be non-negative.

5.3 Social interaction excitation modeling

The social characteristics of user behaviors is important. Through social interactions on social networks, people influence others, and in turn, is influenced by others. To explore social networks influence on user behaviors, we integrate social connections and friends' preferences to model social interactions. Considering the varying importance of social interactions over time, a time-aware strategy is applied. Firstly, we preserve the structure of user's social connections and measure social similarity between user and his friends, which quantize the contribution of friends' preference. Inspired by the social theory, more mutual friends reflect more smaller distance in social networks and more similar preferences in physical world. Jaccard coefficient is a popular way to measure of similarity for two sample sets, by comparing members for two sets with common elements, yielding a value from 0 to 1. Thus, we leverage Jaccard coefficient to measure similarity among users in social network, using Eq. 12. $\Gamma(\cdot)$ indicates the user's friend set, x and y are users in social networks.

$$JC_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{12}$$

Due to the widespread use of smartphones, the information spreads fast, which enhances the influence of social interaction. To capture evolving changes of social interactions, we leverage the last time friend's intensity (i.e., instantaneous rate of friend's check-in activities) as the friend's preferences. Subsequently, we model social interactions between a user and his friends with the product of social similarity and the friend's preference. Further, we sum up social interaction of all of the user's friend (i.e., all directly connected nodes in social networks), and divide by the number of user's friends. Meanwhile, we leverage a time kernel function for learning time effect, which is $e^{-\omega(t-t_j)}$. Here ω is not less than 0. Formally, we introduce the social interactions into the conditional intensity function of the user's behavior, as given in Eq. 13.

$$\lambda_x^*(t) = \beta^T * L_x(t) + \frac{1}{|Nei|} \sum_{y \subseteq Nei} JC_{xy} * \lambda_y(t_{i-1}) \sum_{t_j < t} e^{-\omega(t-t_j)} \tag{13}$$

Where, $\lambda_y(t_{i-1})$ is the friend's intensity last time, and $|Nei|$ is the directly connect node set of user x excluding the user x. Here the first term aims to learn user's personal preferences, and the second term aims to model social interaction excitations. The parameter β and ω are inferred by the learning algorithm of DGPS.

5.4 Dynamic modeling of user's behaviors

Based on the basis of point process framework [1], conditional intenstiy function can infer conditional density function $f^*(t)$, which is given in Eq. 14.

$$\lambda^*(t) = \frac{f^*(t)}{1 - F^*(t)} = \frac{f^*(t)}{Q^*(t)} \tag{14}$$

where, $F^*(t)$ is cumulative probability function, and it indicates the cumulative probability of a new check-in event before time t since last event. $1 - F^*(t)$ represents no check-in event occurrence from time t to t_{i-1}, which is denoted as $Q^*(t)$ and Eq. 15.

$$Q^*(t) = exp\left(-\int_{t_{i-1}}^{t} \lambda^*(\tau)d\tau\right) \tag{15}$$

Now, we can write the conditional density function of temporal point process as Eq. 16, given on last check-in event time.

$$f^*(t_i|t_{i-1}) = \lambda^*(t_i) * exp(-\int_{t_{i-1}}^{t_i} \lambda^*(\tau)d\tau) \tag{16}$$

After obtaining the conditional density function, we can forecast the probability of a location of user's next check-in activity, using check-in activities historical data. Furthermore, DGPS can calculate the expected time of user's next check-in activities via Eq. 17.

$$t_{i+1} = \int_{t_i}^{\infty} \tau f^*(\tau)d\tau \tag{17}$$

However, this expectation is too difficult to calculate, and we estimate the expected time using the method in [9, 29] as follows.

Algorithm 2 DGPS learning and forecasting algorithm.

Input:
Input User's check-in activities with location features $\{(f_{l1}, t_1), (f_{l2}, t_2), ...(f_{lT}, t_T)\}$;
Social networks;
Output:
Output Forecasting result of individual user check-in event location and time;

for each user u in datasets **do**
  calculate JC similarity using Eq. 12;
end for
initialize β, ω;
while $\mathbf{O}_{\beta,\omega,z,u}$ not converges **do**
  update β by Eq. 24;
  update z by Eq. 25;
  update u by Eq. 26;
  update ω via gradient descent using Eq. 28;
end while
for each user u in datasets **do**
  calculate conditional intensity $f^*(t_{i+1})$ using Eq. 16;
  calculate time expectation t_{i+1} using Eq. 18;
end for
return $f^*(t_{i+1})$, t_{i+1};

$$t_{i+1} = \frac{1}{T}\sum_{i=1}^{T} t_i^{n+1} \tag{18}$$

where, t_i^{n+1} is the next expected time of i-th time interval.

1 Calculate the expected time interval $t_1^{n+1}, ...t_T^{n+1}$ by Ogata's thinning algorithm;
2 Calculate the average expected time in history as the estimating time via Eq. 18.

Given user's check-in historical activity data $\{(f_{l1}, t_1), (f_{l2}, t_2), ...(f_{lT}, t_T)\}$ and conditional intensity function $\lambda^*_{\beta,\omega}(t)$ (the subscripts indicate conditional intensity function, which is related to parameters β, ω). Formally, we write the likelihood function and log-likelihood function in Eqs. 19 and 20, respectively. In the following section, we will introduce how to learning the two parameters: β and ω.

$$\begin{aligned}\mathcal{L}(\beta,\omega) &= \prod_{i=1}^{T} f^*(t_i|t_{i-1}) \\ &= \prod_{i=1}^{T} \lambda^*_{\beta,\omega}(t_i)exp\left(-\int_{t_{i-1}}^{t_i} \lambda^*_{\beta,\omega}(\tau)d\tau\right)\end{aligned} \tag{19}$$

$$\begin{aligned}\ln(\mathcal{L}|\beta,\omega) &= \ln\left(\prod_{i=1}^{T} \lambda^*_{\beta,\omega}(t_i)exp\left(-\int_{t_{i-1}}^{t_i} \lambda^*_{\beta,\omega}(\tau)d\tau\right)\right) \\ &= \sum_{i=1}^{T} \ln(\lambda^*_{\beta,\omega}(t_i)) - \int_0^T \lambda^*_{\beta,\omega}(\tau)d\tau\end{aligned} \tag{20}$$

5.5 Parameter learning

To learn model parameters, we minimize negative log-likelihood function of Eq. 20. Meanwhile, we add the sparsity regularization $\|\beta\|_1$ into log-likelihood function for parameter control, and finally, we has Eq. 21. Due to non-differentiable problem arising from ℓ_1 norm, we leverage alternating direction method of multipliers framework to solve the optimization problem of objective function, which is a popular way for addressing the problem of non-differentiable [4]. Based on ADMM, we break the optimization problem of objective function into some sub-problems, each of which is easier to handle.

$$\begin{aligned}\mathbf{O}_{\beta,\omega} &= -\left(\sum_{i=1}^{T} \ln(\beta^T * L(t_i) + \frac{1}{|Nei|}\sum_{y\subseteq Nei} JC * \lambda_y(t_{i-1})\sum_{t_j<t} e^{-\omega(t-t_j)}\right) \\ &\quad -\int_o^T (\beta^T * L(\tau) + \frac{1}{|Nei|}\sum_{y\subseteq Nei} JC * \lambda_y(\tau_{i-1})\sum_{t_j<t} e^{-\omega(\tau-t_j)}d\tau) + \gamma\|\beta\|_1\end{aligned} \tag{21}$$

Under ADMM framework, we rewrite an equivalent form of the objective function (Eq. 21) by introducing an auxiliary parameter z, as in Eq. 22.

$$\mathbf{O}_{\beta,\omega,z} = \mathbf{O}_{\beta,\omega} + \gamma\|z\|_1 \quad \text{s.t.} \quad \beta = z \tag{22}$$

Then we formulate the augmented Lagrangian function $\mathbf{O}_{\beta,\omega,z,u}$, as Eq. 23.

$$\mathbf{O}_{\beta,\omega,z,u} = \mathbf{O}_{\beta,\omega} + \gamma\|z\|_1 + \rho u^T(\beta - z) + \frac{\rho}{2}\|\beta - z\|_2^2 \tag{23}$$

We convert $\mathbf{O}_{\beta,\mathbf{s}\omega,\mathbf{z},\mathbf{u}}$ to three sub-problems involving parameters β, z, u, denoted by Eq. 24 to Eq. 26.

$$\beta^{k+1} := \arg\min_{\beta} \left(\mathbf{O}_{\beta^k,\omega} + \frac{\rho}{2}\|\beta^k - z^k + u^k\|_2^2\right) \tag{24}$$

$$z^{k+1} := \arg\min_{z} \left(\gamma \| z^k \|_1 + \frac{\rho}{2} \| z^k - \beta^{k+1} - u^k \|_2^2 \right) \tag{25}$$

$$u^{k+1} := u^k + \rho \left(\beta^{k+1} - z^{k+1} \right) \tag{26}$$

We adopt gradient descent method for updating β in Eq. 24, and gradient of β is Eq. 27. $\mathbf{O}_\beta$ is related item of β in the objective function Eq. 21. The updating method of z uses the soft thresholding [7].

$$\frac{\partial \mathbf{O}_\beta}{\partial \beta} = -\left(\sum_{i=1}^{T} \frac{L(t_i)}{\beta^T * L(t) + \frac{1}{|Nei|} \sum_{y \subseteq Nei} JC * \lambda_y(t_{i-1}) \sum_{t_j < t} e^{-\omega(t-t_j)}} \right. \\ \left. - \int_0^T L(s) ds \right) \tag{27}$$

Similarly, we update ω by gradient descent method, Eq. 28 is ω gradient. $\mathbf{O}_\omega$ is related item of ω in the objective function Eq. 21.

$$\frac{\partial \mathbf{O}_\omega}{\partial \omega} = \sum_{i=1}^{T} \frac{\sum_{t_j < t_i} \frac{1}{|Nei|} \sum_{y \subseteq Nei} JC * \lambda_y(t_{i-1}) e^{-w(t_i - t_j)(t_j - t_i)}}{\beta^T * L(t_i) + \frac{1}{|Nei|} \sum_{y \subseteq Nei} JC * \lambda_y(t_{i-1}) \sum_{t_j < t} e^{-\omega(t-t_j)}} \\ - \frac{\sum_{i=1}^{T} \frac{1}{|Nei|} \sum_{y \subseteq Nei} JC * \lambda_y(t_{i-1}) e^{-\omega(T-t_i)} (T - t_i)\omega - (1 - e^{-\omega(T-t_i)})}{\omega^2} \tag{28}$$

We rank $f^*(t_{i+1})$ result for forecasting the next check-in location. Meanwhile, we estimate the expected time of next check-in activity by Eq. 18. Finally, we give the pseudocode of DGPS in Algorithm 2.

5.6 Model complexity

We divide DGPS into two steps for analysing time complexity. The first stage is geo-aware location feature engineering process. The time complexity of non-negative matrix is $o((m+n)k)$, where m is the number of user, n is the number of location, and k is the number of latent factors. The complexity of semantic features and dynamic features extracting process is $o(n)$. The time complexity of first stage is thus $o((m+n)k) + o(n)$. The second stage is DGPS model learning algorithm. Given a check-in dataset, and each user's the number of average check-ins event is N. The length of time lines is T. DGPS learning algorithm leverages maximum likelihood estimation and alternative direction method of multipliers framework to solve problem of parameter optimization. According to the complexity calculation methods in [31, 41], the complexity of DGPS learning algorithm is $o(mN^2T)$. Therefore, the total time complexity of DGPS is approximated to $o((m+n)k) + o(n) + o(mN^2T)$.

6 Experiments

In this study, we aim to accurately predict location and time of user's next check-in activity. To comprehensively evaluate the performance of DGPS, we conducted extensive experiments on several real-world data sets from Brightkite, Gowalla, Foursquare check-in data sets [5, 25, 45]. Meanwhile, we compared DGPS with some state-of-art methods. An introduction to and the parameter settings of these models is provided in Section 6.5. All the experiments were performed on a computer with four cores (3.5 GHz CPU) and 32 GB of RAM and a single NVIDIA GPU with 8 GB memory. Details of datasets and experiments settings are given in the following sections.

6.1 Experiments settings

In this study, we focus on three check-ins datasets. Each check-in record contains the user ID, check-in time, location ID and the latitude and longitude of the check-in location. Details of data sets are as follows:

- **BrightKite** check-in dataset was provided by a location-based social network service provider. The social network of data set consists of 58,228 nodes and 214,078 edges. The data set also collected a total of 4,491,143 check-ins of these users over the period of Apr. 2008 - Oct. 2010. In order to obtain the semantic label of nearby place of the check-in location, we divide the data set into five states (or city) in USA, including California, Colorado, New York, Texas and Washington states (or cities).
- **Gowalla** check-in dataset has collected a total of 6,442,890 check-ins over the period of Feb. 2009 - Oct. 2010. We selected New York city as a representative data set.
- **Foursquare** check-in data set has collected a total of 573,703 check-ins over the period of Apr. 2012 - Feb. 2013. We selected Tokyo city as a representative data set.

In the data preprocessing, we remove the users who have less than ten check-ins. After that, Table 3 further lists the statistics of check-in datasets in corresponding states (or cities). We split these datasets into 70% for training and 30% for test, in a chronological order. The time interval decides training and prediction time granularity. We set the time interval as 7 days in this study. The other parameters are set as follows: $k = 90$, $\gamma = 0.1$ in California, New York(B), Texas and Washington datasets, $\gamma = 0.5$ in Colorado datasets, and $k = 90$, $\gamma = 0.1$ in New York(G), Tokyo data sets. The effects of these parameters are further investigated in Section 6.6.

6.2 Evaluation metrics

To quantitatively study the comprehensive performance of DGPS, we adopted the following performance metrics:

- *Prediction performances.* We evaluate the performance of DGPS in term of precision (including Precision@p, $p = 1, 5, 10, 20$), Recall at predicted length 20, F1 score, rank (average precision rank, APR) and time deviation.

Table 3 The statistic of data sets(Check-ins are the number of check-in record, Locations are the number of unique locations, Users are the number of unique users, Socials are the number of edges in social networks, New York(B) is BrightKite data set, New York(G) is Gowalla data set)

Data sets	California	Colorado	New York(B)	Texas	Washington	New York(G)	Tokyo
Check-ins	521622	182413	138537	221694	126047	47310	390118
Locations	54581	27128	24164	35228	18191	2249	10647
Users	6042	2075	1174	3214	1708	1327	5079
Socials	44914	14932	14928	19798	7698	16845	37092

- *Dynamic performance.* We simulate the check-in intensity of the selected user using check-in historical data for revealing the dynamic performances of DGPS. Meanwhile, we also show the semantic label of location of the selected user.
- *Sensitivity.* We test the sensitivity of the prediction performance to parameter variation.

F1 score. Both the accuracy and recall are considered into the F1 score, which is the harmonic average of the accuracy and recall value, as Eq. 29.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{29}$$

APR (average percentile rank). The percentile rank of prediction for location l_j is Eq. 30 [34].

$$PR = \frac{n - rank(l_j) + 1}{n} \tag{30}$$

where n is the number of locations, $rank(l_j)$ is the position of location l_j in the predicted list. APR is average of PR.

Time-dev (time deviation). We calculate average differences between true time and predicted time using Eq. 31.

$$\text{Time-dev} = \frac{|T_{real} - T_{pre}|}{T_{real}} \tag{31}$$

where T_{real} is the true time of user check-in activity, T_{pre} is predicted time of user check-in activity.

6.3 Prediction performance of DGPS

In this section, we report the prediction performance of DGPS. Table 4 summarizes the prediction performance in terms of the different evaluation measures on the data sets of several states (or cities). From Table 4, we can observe that DGPS achieves a good prediction performance, especially for the APR and Time-dev. In addition, based on experiments of prediction performance, the following conclusions have been drawn: (1) Data sparsity may has some effects to DGPS. For example, the performance on Texas dataset is less than the other datasets (sparseness is 2×10^{-5}). In Texas dataset, time intervals of most successive check-in activities are more than one month. Extreme sparsity problem leads to DGPS

Table 4 The forecasting performance of DGPS on different data sets.(New York(B) is BrightKite dataset, New York(G) is Gowalla dataset.)

Data sets	Precision@1	Precision@5	Precision@10	Precision@20	Recall	F1 score	APR	Time-dev
California	0.102	0.139	0.208	0.237	0.204	0.219	0.897	0.086
Colorado	0.087	0.128	0.171	0.187	0.175	0.181	0.862	0.091
New York(B)	0.075	0.179	0.201	0.229	0.239	0.234	0.883	0.087
Texas	0.089	0.097	0.150	0.178	0.191	0.184	0.797	0.104
Washington	0.109	0.142	0.178	0.227	0.259	0.242	0.892	0.111
New York(G)	0.067	0.103	0.128	0.147	0.162	0.154	0.715	0.177
Tokyo	0.082	0.142	0.161	0.169	0.181	0.175	0.857	0.113

that cannot well capture the user's preferences. (2) Richer social relations are benefits to DGPS for achieving better prediction performance. For example, average social relations of per user are 7.4, 7.2, 12.7, 6.1 and 4.5 in California, Colorado, New York(B), Texas and Washington (i.e. divide the total number of social relations to the total number of users), respectively. The prediction performance on California, Colorado and New York(B) is better than Texas. The experimental results demonstrate the successes of social interaction excitation modeling. (3) The number of average check-in record of each location has a significant effect to the prediction performance of DGPS. Such as, the user in California and New York(B) have more check-in records of per user than the user in Texas and Washington. The performance of DGPS on California and New York(B) data set obtains better results than Texas and Washington. But the people in Colorado is a little different. The potential reason is that users in this city like to explore new locations. (4) For the Tokyo population, the performance of DGPS on the Foursquare data set was similar to that achieved by BrightKite, but for the New York population the performance of DGPS on the Gowalla dataset was poor, likely due to data sparsity. The average number of check-ins of per user was less than 36 in the New York(G) data set, the lowest among all the datasets.

As all location features are essential to learn user preferences and improve the performance of DGPS, we study their impacts in this experiment. To further explore the impact of different features in DGPS, we compare DGPS with its three variants using different combination of features. Table 5 shows the variant using the combination of different features. DGPS-SL, DGPS-SD and DGPS-LD lack one specific feature, and DGPS-SLD includes all features.

Tables 6 and 7 show the prediction performance on California and Colorado data sets. From both tables, we observe that DGPS-SLD achieves the best performance than the three variant on both data sets, indicating the benefits brought by all features. In addition, DGPS-LD outperforms than DGPS-SL and DGPS-SD, and the performance of DGPS-SL is close to DGPS-SD. The results illustrate the contribution to DGPS among different features are also not the same. We conclude latent features and dynamic features are more important than semantic features. But if DGPS lacks any specific features, it cannot achieve good performance.

6.4 Dynamic analysis

To further evaluate the dynamic performance of DGPS on user's behavior modeling, we carry on a case study to compare simulated intensity of check-in activities with real visiting variation. We select three representative users from New York, Texas and Washington, and we calculate variation of simulated intensity of each user, respectively. Meanwhile, we collect the number of check-ins on each time unit for corresponding user, that is ground truth. Then we normalize both simulated results and ground truth, and plot both of them with time

Table 5 Different features of DGPS

	Semantic features	Latent features	Dynamic features
DGPS-SL	•	•	
DGPS-SD	•		•
DGPS-LD		•	•
DGPS-SLD	•	•	•

Table 6 The impact of different features of DGPS on prediction performances in the California data set

Features	Precision@1	Precision@5	Precision@10	Precision@20	Recall	F1 score	APR	Time-dev
DGPS-SL	0.051	0.091	0.113	0.130	0.173	0.148	0.720	0.205
DGPS-SD	0.060	0.103	0.147	0.167	0.178	0.172	0.773	0.104
DGPS-LD	0.084	0.115	0.156	0.172	0.181	0.176	0.816	0.095
DGPS-SLD	0.102	0.139	0.208	0.237	0.204	0.219	0.897	0.086

variation. The left figures in Figs. 6, 7 and 8 are the comparison between simulated intensity and ground truth. From these figures we can see that, although the value of simulated intensity is higher than the real value, simulated result of DGPS can well fit the instantaneous rate of real check-in event with time variation. Meanwhile, we draw the wordcloud of semantic label of user visited venue. Excluding the common place such as store restaurant, worldcloud of semantic label of visited location points out the user's life preferences. Figure 6b shows that a user from New York more likely be a women since the user's check-in regions have many cloth store, show store, bar and night club, etc. The user from Texas in Fig. 7b may be a doctor, since he always check in locations with doctor labels. Figure 8b shows the user from Washington has check-in activities in many venues, and thus we infer that he might have a colourful life in his leisure time.

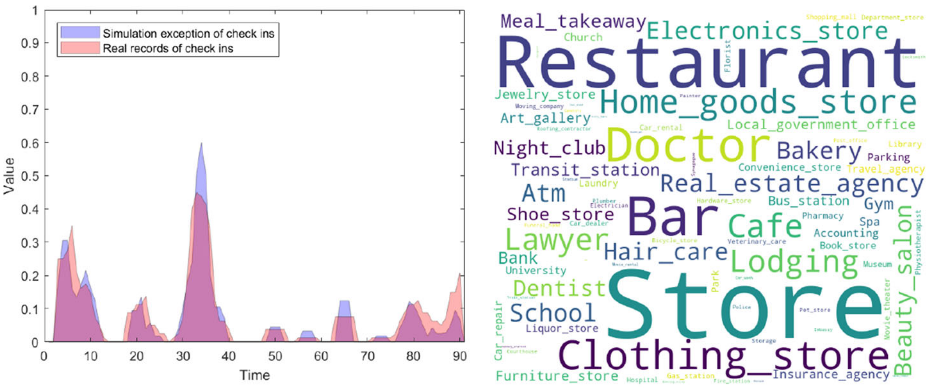
6.5 Comparisons with other algorithms

To further demonstrate the benefits of DGPS, we compare DGPS with seven baselines.

- **RSTE.** Recommend with Social Trust Ensemble[26] considers both user's taste and their trusted friends' favors into a probabilistic factor analysis model. Without any rating information in our datasets, we use the frequencies matrix X replace the rating matrix R. Meanwhile, directly connected nodes simulate the trusted friends sets in RSTE model.
- **GeoMF.** GeoMF introduces spatial clustering phenomenon into weight matrix factorization model for POIs recommendation [21]. Specifically, GeoMF augments user's activity areas and POI's geographical influence areas into user's and POI's matrices, respectively. We set the number of latent factors as 30, and the other parameters are fine tune suggested by the paper.
- **HP.** Hawkes process is a classical temporal point process, and we fit a self-excitation Hawkes process using the intensity function in [39], and the parameters are initialized randomly.
- **W3.** W4 (who, when, where, and what) is a probability model fusing user's profile, location, twitter and temporal information etc. [48]. It can predict user's location

Table 7 The impact of different features on DGPS prediction performances in the Colorado data set

Features	Precision@1	Precision@5	Precision@10	Precision@20	Recall	F1 score	APR	Time-dev
DGPS-SL	0.058	0.093	0.133	0.141	0.151	0.146	0.733	0.143
DGPS-SD	0.071	0.096	0.151	0.161	0.153	0.157	0.747	0.121
DGPS-LD	0.081	0.114	0.162	0.170	0.163	0.166	0.790	0.109
DGPS-SLD	0.087	0.128	0.171	0.187	0.175	0.181	0.862	0.091

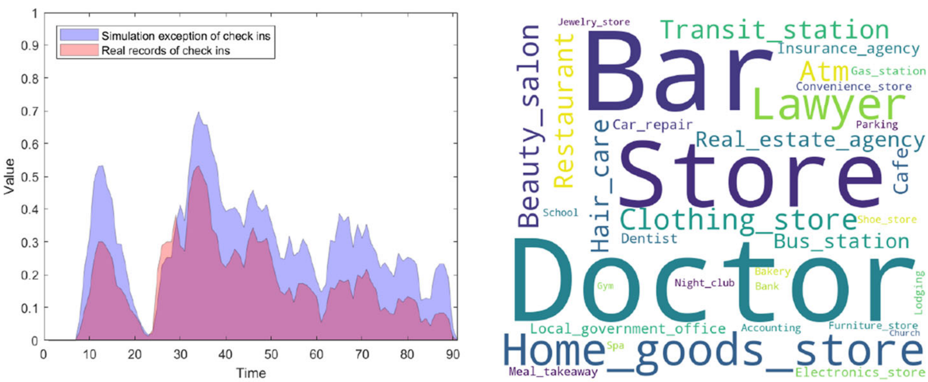


(a) (b)

Fig. 6 Case study in New York (user ID is 293). Figure **a** shows comparison between simulated intensity and real visiting frequencies. Figure **b** shows the wordcloud of semantic label of visited location of the user

via his twitter and check-in historical data. Since our data set does not have the text information, we remove the attribute of what, then W4 transforms to W3.

- **LSTM.** Long Short-Term Memory network is a recurrent neural network in the area of deep learning. LSTM can remember the long effect and short-term influence via a unique cell (including input, forget and output gate), which are well-suited to making prediction based on time series data. In this study, we use LSTM as the representation of deep learning methods instead of GRU (Gated recurrent unit). GRU always achieves better performance than LSTM in NLP area instead of sequence modeling. In this study, after extracting location features, we build a location-feature sequence. Then we use a standard LSTM [14] to train a sequence prediction model under PyTorch framework. Parameters are fine tuned to yield the best result.



(a) (b)

Fig. 7 Case study in Texas (user ID is 689). Figure **a** shows comparison between simulated intensity and real visiting frequencies. Figure **b** shows the wordcloud of semantic label of visited location of the user

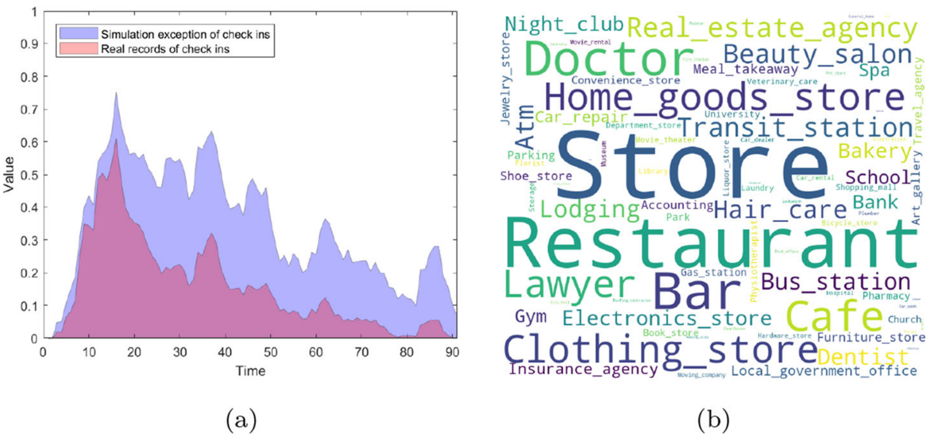


Fig. 8 Case study in Washington (user ID is 1974). Figure **a** shows comparison between simulated intensity and real visiting frequencies. Figure **b** shows the wordcloud of semantic label of visited location of the user

- **MFM.** Most Frequent Model is a simple baseline to making prediction based on user's frequencies to a location. We rank the frequencies of the user's frequency based on his check-in history, and assign the high value locations as the predicted result.
- **PMF.** Probability matrix factorization is common method based on matrix factorization, which is widely used in recommendation tasks [27].

Figures 9 and 10 summarize the prediction performance in terms of different measures on real-world datasets of California and Colorado states. The other algorithms cannot predict the time of user' check-in activity, so the measure of the time-dev is only provided by DGPS. From these figures, we can observe that DGPS achieves the best results. Good performance of DGPS may be due to following reasons: 1) Under DGPS framework, semantic

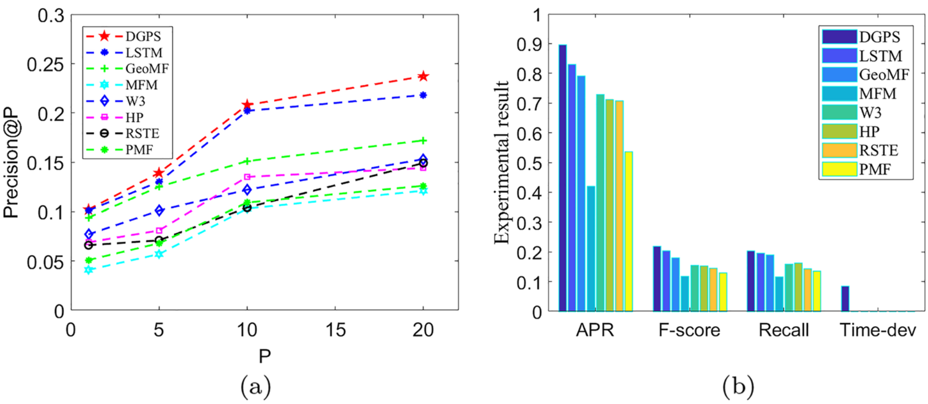


Fig. 9 Figure **a** and **b** compare the performance of other algorithms with DGPS on California dataset. Figure **a** is Precision@P performance, P is the predicted length. Figure **b** is experiment result of the APR, Recall, F-score and Time-dev. Other algorithms do not provide capacity of forecasting time, so we set results to zero

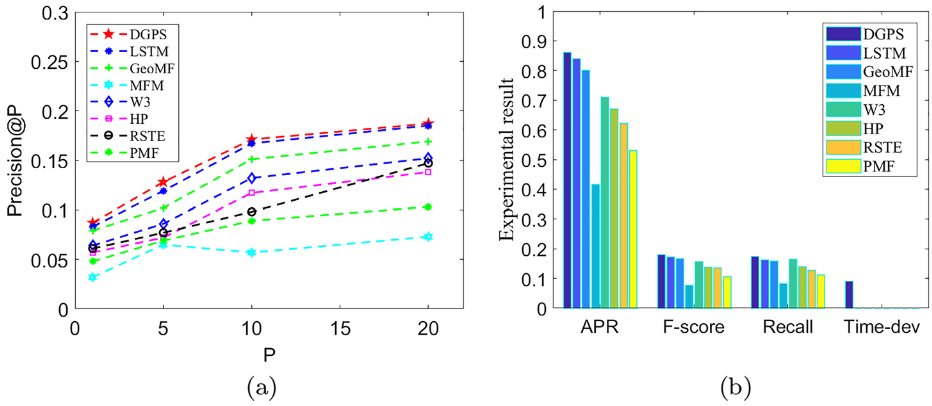


Fig. 10 Figure **a** and **b** compare the performance of other algorithms with DGPS on Colorado dataset. The meaning of figure is same as Fig. 9

Table 8 Sensitivity of hidden features parameter k on California dataset

k	Precision@1	Precision@5	Precision@10	Precision@20	Recall	F1 score	APR	Time-dev
10	0.061	0.117	0.156	0.162	0.173	0.167	0.810	0.115
30	0.067	0.118	0.162	0.166	0.175	0.170	0.810	0.109
50	0.072	0.121	0.167	0.187	0.192	0.189	0.812	0.108
70	0.101	0.130	0.202	0.218	0.223	0.220	0.891	0.091
90	0.102	0.139	0.208	0.237	0.204	0.219	0.897	0.086

Table 9 Sensitivity of model parameter γ on California dataset

γ	Precision@1	Precision@5	Precision@10	Precision@20	Recall	F1 score	APR	Time-dev
0.01	0.097	0.127	0.204	0.228	0.202	0.214	0.827	0.093
0.1	0.102	0.139	0.208	0.237	0.204	0.219	0.897	0.086
0.5	0.097	0.131	0.192	0.224	0.204	0.213	0.852	0.091
1	0.093	0.128	0.183	0.190	0.187	0.188	0.831	0.091
10	0.082	0.119	0.141	0.162	0.174	0.168	0.811	0.102

Table 10 Sensitivity of hidden features parameter k on Colorado dataset

k	Precision@1	Precision@5	Precision@10	Precision@20	Recall	F1 score	APR	Time-dev
10	0.057	0.091	0.127	0.145	0.158	0.151	0.783	0.111
30	0.061	0.103	0.134	0.157	0.161	0.159	0.790	0.105
50	0.079	0.117	0.161	0.179	0.165	0.172	0.851	0.094
70	0.085	0.128	0.168	0.181	0.169	0.174	0.857	0.094
90	0.087	0.128	0.171	0.187	0.175	0.181	0.862	0.091

Table 11 Sensitivity of model parameter γ on Colorado dataset

γ	Precision@1	Precision@5	Precision@10	Precision@20	Recall	F1 score	APR	Time-dev
0.01	0.074	0.117	0.158	0.167	0.172	0.169	0.851	0.103
0.1	0.081	0.124	0.171	0.181	0.177	0.179	0.855	0.088
0.5	0.087	0.128	0.171	0.187	0.175	0.181	0.862	0.091
1	0.070	0.121	0.169	0.176	0.173	0.175	0.860	0.099
10	0.072	0.114	0.167	0.171	0.165	0.168	0.853	0.101

features, latent features and dynamic features are fused into the geo-aware location features. Especially, fusing latent features is the main reason to gain better performance than GeoMF, RSTE, and PMF. Relying on geo-aware location features, DGPS can well capture user's personal preferences and gains better performance than traditional preference learning model. 2) DGPS integrates social connections and evolving friends' preference into a social interaction excitation modeling. Although time-dependent dynamic model face with overly exacting challenge of data sparsity than traditional static model, the social interaction excitation modeling in DGPS is helpful to alleviate data sparsity and improve performances. 3) More importantly, we observe DGPS and LSTM always gain better results than other methods on both datasets. The reason behind this experimental result is that DGPS can capture the dynamic mobility mechanism of user's activities on LBSNs, and LSTM can learn context information in check-in activity. Although LSTM can access approximate results to DGPS on Colorado dataset, LSTM lacks the interpretation of user's check-in behaviors and cannot intuitively reveal the influence factors of dynamic mobility mechanism of user's check-in behaviors.

6.6 Sensitivity to parameters

In this section, we perform sensitivity analysis of DGPS by varing the values of different parameters on the datasets of California and Colorado, including the number of latent factors k, the coefficient of the ℓ_1 norm γ. We vary k from 10 to 90 with stepsize of 20, and γ from 0.01 to 10. Tables 8, 9, 10 and 11 show the DGPS's performance with varying values of different parameters.

From these tables, we can see that the model performance grows up with increasing k, but it will not better when k gets a reasonable value, such as $k = 70$ on California dataset and $k = 50$ on Colorado dataset. The performance of DGPS is quite robust when γ not more than 1 on both two datasets. In general, DGPS is not too sensitive to parameters.

7 Discussion and conclusion

In this paper, we propose a novel dynamic model to learn user's check-in behaviors on LBSNs, in which the dynamic model jointly integrates geo-aware user preferences learning and social interaction excitation modeling through a proposed conditional intensity function of temporal point process. In particular, to explore the effect of geographical influence to user preferences, DGPS extracts the geo-aware features of a location, which include semantic features, latent features and dynamic features. Meanwhile, we integrate social connections, friends' preferences, and time kernel function for modeling social interaction excitation. Furthermore, DGPS simulates the temporal point process to learn dynamic mobility mechanism of evolving user behaviors. Another attractive property of DGPS is that it provides a way to forecast the location as well as time for future check-in activity. Through comprehensive experiments, we demonstrate that DGPS outperforms many state-of-art models on LBSNs, and it also provides an intuitive way to analyze the dynamic mobility mechanism of user's check-in behaviors.

Acknowledgements This work was supported in part China Scholarship Council (201806070074), in part by the Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161062), in part by National Key Research and Development Program under Grant (2016YFB0502300), in part by Key Research Plan for State Comission of Science Technology of China (2018YFC0807501), in part by Sichuan Science and Technology Program (No.2017JZ0031,2018JY0578,2018JY0067), and Chengdu Science and Technology Bureau Program (2018-YF09-00051-SN).

References

1. Aalen O, Borgan O, Gjessing H (2008) Survival and event history analysis: A process point of view. Springer Science & Business Media
2. Bao J, Zheng Y, Wilkie D, Mokbel M (2015) Recommendations in location-based social networks: A survey. GeoInformatica 19(3):525–565
3. Bhargava P, Phan T, Zhou J, Lee J (2015) Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In: Proceedings of the 24th international conference on World Wide Web, pp 130–140. ACM
4. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends®in Machine learning 3(1):1–122
5. Cho E, Myers SA, Leskovec J (2011) 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, pp 1082–1090. ACM
6. Daley DJ, Vere-Jones D (2007) An introduction to the theory of point processes: volume II: General theory and structure. Springer Science & Business Media
7. Donoho DL (1995) De-noising by soft-thresholding. IEEE Trans Inf Theory 41(3):613–627
8. Du N, Dai H, Trivedi R, Upadhyay U, Gomez-Rodriguez M, Song L (2016) 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, pp1555–1564. ACM
9. Du N, Wang Y, He N, Sun J, Song L (2015) Time-sensitive recommendation from recurrent user activities. In: Advances in neural information processing systems, pp 3492–3500
10. Ferraz Costa A, Yamaguchi Y, Juci Machado Traina A, Traina C Jr, Faloutsos C (2015) Rsc: Mining and modeling temporal activity in social media. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 269–278. ACM
11. Gambs S, Killijian MO, del Prado Cortez MN (2012) Next place prediction using mobility markov chains. In: Proceedings of the 1st workshop on measurement, privacy, and mobility, pp 3. ACM
12. Gao H, Tang J, Liu H (2012) Exploring social-historical ties on location-based social networks. In: ICWSM, pp 114–121
13. Gao H, Tang J, Liu H (2012) gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. In: Proceedings of the 21st ACM international conference on Information and knowledge management, pp 1582–1586. ACM
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
15. Jia Y, Wang Y, Jin X, Cheng X (2016) Location prediction: A temporal-spatial bayesian model. ACM Trans Intell Syst Technol (TIST) 7(3):31
16. Lee H, Yoo J, Choi S (2010) Semi-supervised nonnegative matrix factorization. IEEE Signal Process Lett 17(1):4–7
17. Li M, Westerholt R, Fan H, Zipf A (2018) Assessing spatiotemporal predictability of lbsn: A case study of three foursquare datasets. GeoInformatica 22(3):541–561

18. Li X, Cong G, Li XL, Pham TAN, Krishnaswamy S (2015) Rank-geofm: a ranking based geographical factorization method for point of interest recommendation. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp 433–442. ACM
19. Lian D, Xie X, Zheng VW, Yuan NJ, Zhang F, Chen E (2015) Cepr: A collaborative exploration and periodically returning model for location prediction. ACM Trans Intell Syst Technol (TIST) 6(1):8
20. Lian D, Zhang Z, Ge Y, Zhang F, Yuan NJ, Xie X (2016) Regularized content-aware tensor factorization meets temporal-aware location recommendation. In: 2016 IEEE 16th international conference on data mining (ICDM), pp 1029–1034. IEEE
21. Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y (2014) Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 831–840. ACM
22. Liu B, Fu Y, Yao Z, Xiong H (2013) Learning geographical preferences for point-of-interest recommendation. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1043–1051. ACM
23. Liu Q, Wu S, Wang L, Tan T (2016) Predicting the next location: A recurrent model with spatial and temporal contexts. In: AAAI, pp 194–200
24. Liu X, Yan J, Xiao S, Wang X, Zha H, Chu SM (2017) On predictive patent valuation: Forecasting patent citations and their types. In: AAAI, pp 1438–1444
25. Liu Y, Liu C, Liu B, Qu M, Xiong H (2016) Unified point-of-interest recommendation with temporal interval assessment. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1015–1024. ACM
26. Ma H, King I, Lyu MR (2009) Learning to recommend with social trust ensemble. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp 203–210. ACM
27. Mnih A, Salakhutdinov RR (2008) Probabilistic matrix factorization. In: Advances in neural information processing systems, pp 1257–1264
28. Monreale A, Pinelli F, Trasarti R, Giannotti F (2009) Wherenext: A location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 637–646. ACM
29. Ogata Y (1981) On lewis' simulation method for point processes. IEEE Trans Inf Theory 27(1):23–31
30. Ogata Y (1988) Statistical models for earthquake occurrences and residual analysis for point processes. J Am Stat Assoc 83(401):9–27
31. Rizoiu MA, Lee Y, Mishra S, Xie L (2017) A tutorial on hawkes processes for events in social media. arXiv:1708.06401
32. Shao J, Han Z, Yang Q, Zhou T (2015) Community detection based on distance dynamics. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, Sydney, NSW, Australia, August 10-13, 2015, pp 1075–1084
33. Song C, Koren T, Wang P, Barabási AL (2010) Modelling the scaling properties of human mobility. Nat Phys 6(10):818
34. Wang Y, Yuan NJ, Lian D, Xu L, Xie X, Chen E, Rui Y (2015) Regularity and conformity: Location prediction using heterogeneous mobility data. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1275–1284. ACM
35. Wu R, Luo G, Shao J, Tian L, Peng C (2018) Location prediction on trajectory data: A review. Big Data Mining and Analytics 1(2):108–127
36. Wu R, Luo G, Yang Q, Shao J (2018) Learning individual moving preference and social interaction for location prediction. IEEE Access 6:10675–10687
37. Xiao S, Farajtabar M, Ye X, Yan J, Song L, Zha H (2017) Wasserstein learning of deep generative point process models. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R. (eds) Advances in neural information processing systems 30. Curran Associates, Inc, New York, pp 3247–3257
38. Xiao S, Yan J, Li C, Jin B, Wang X, Yang X, Chu SM, Zha H (2016) On modeling and predicting individual paper citation count over time. In: IJCAI, pp 2676–2682
39. Xiao S, Yan J, Yang X, Zha H, Chu SM (2017) Modeling the intensity function of point process via recurrent neural networks. In: AAAI, vol 17, pp 1597–1603
40. Xu B, Ding Z, Chen H (2017) Recommending locations based on users' periodic behaviors. Mob Inf Syst 2017:1–9
41. Xu H, Farajtabar M, Zha H (2016) Learning granger causality for hawkes processes. In: International conference on machine learning, pp 1717–1726

42. Xu T, Zhong H, Zhu H, Xiong H, Chen E, Liu G (2015) Exploring the impact of dynamic mutual influence on social event participation. In: Proceedings of the 2015 SIAM international conference on data mining, pp 262–270. SIAM
43. Xu T, Zhu H, Zhong H, Liu G, Xiong H, Chen E (2018) Exploiting the dynamic mutual influence for predicting social event participation. IEEE Trans Knowl Data Eng 31(6):1122–1135
44. Xu Z, Cai Z, Li J, Hong G (2017) Location-privacy-aware review publication mechanism for local business service systems. In: IEEE INFOCOM 2017 - IEEE conference on computer communications, pp 1–9
45. Yang D, Zhang D, Zheng VW, Yu Z (2014) Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. IEEE Trans Syst Man Cybern Syst 45(1):129–142
46. Yin H, Wang W, Wang H, Chen L, Zhou X (2017) Spatial-aware hierarchical collaborative deep learning for poi recommendation. IEEE Trans Knowl Data Eng 29(11):2537–2551
47. Yin H, Zhou X, Cui B, Wang H, Zheng K, Nguyen QVH (2016) Adapting to user interest drift for poi recommendation. IEEE Trans Knowl Data Eng 28(10):2566–2581
48. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) 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, pp 605–613. ACM
49. Zhang W, Wang J, Feng W (2013) Combining latent factor model with location features for event-based group recommendation. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, pp 910–918. ACM
50. Zhao L, Wang J, Chen F, Lu CT, Ramakrishnan N (2017) Spatial event forecasting in social media with geographically hierarchical regularization. Proc IEEE 105(10):1953–1970
51. Zheng Y, Capra L, Wolfson O, Yang H (2014) Urban computing: Concepts, methodologies, and applications. ACM Trans Intell Syst Technol (TIST) 5(3):38

Ruizhi Wu received the B.S. degree from Hangzhou Dianzi University in 2008. He is currently pursuing the Ph.D. degree in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His interests include spatialtemporal data mining, data mining, machine learning.

Guangchun Luo received the B.S. degree, M.S. degrees and Ph.D. degrees in computer science from University of Electronic Science and Technology of China, Chengdu, China, in 1995, 1999 and 2004, respectively. He is currently a Professor of computer science at the University of Electronic Science and Technology of China, Chengdu, China. His research interests include computer networking, cloud computing, big data.

Qi Jin is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include heterogeneous data fusion and network data analysis.

Junming Shao received his Ph.D. degree with highest honer (Summa Cum Laude at the University of Munich, Germany, in 2011. He became the Alexander von Humboldt Fellow in 2012. Currently, he is professor of School of Computer Science and Engineering at the University of Electronic Science and Technology of China. His research interests include data mining and neuroimaging. He not only published papers on top-level data mining conferences like KDD, ICDM, SDM (two of those papers have won the Best Paper Award), but also published data mining-related interdisciplinary work in leading journals including Brain, Neurobiology of Aging, and Water Research.

Chang-Tien Lu received the MS degree in computer science from the Georgia Institute of Technology, in 1996 and the PhD degree in computer science from the University of Minnesota, in 2001. He is a professor at the Department of Computer Science, Virginia Tech. His research interests include spatial databases, data mining, geographic information systems, and intelligent transportation systems.

Affiliations

Ruizhi Wu[1] · Guangchun Luo[2] · Qi Jin[1,3] · Junming Shao[1] · Chang-Tien Lu[4]

Ruizhi Wu
ruizhiwu@std.uestc.edu.cn

Qi Jin
qijin@std.uestc.edu.cn

Junming Shao
junmshao@uestc.edu.cn

Chang-Tien Lu
ctlu@vt.edu

1 School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

2 School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China

3 Chengdu Municipal Public Security Bureau, Chengdu, China

4 Department of Computer Science, Virginia Tech, Falls Church, VA, USA