GeoInformatica (2020) 24:477–504
https://doi.org/10.1007/s10707-019-00360-3

Behavior-based location recommendation on location-based social networks

Seyyed Mohammadreza Rahimi [1] · Behrouz Far [2] · Xin Wang [1]

Received: 5 July 2018 / Revised: 11 April 2019 / Accepted: 23 April 2019 /
Published online: 25 May 2019

Abstract

Location recommendation methods on location-based social networks (LBSN) discover the locational preference of users along with their spatial movement patterns from users' check-ins and provide users with recommendations of unvisited places. The growing popularity of LBSNs and abundance of shared location information has made location recommendation an active research area in the recent years. However, the existing methods suffer from one or more deficiencies such as data sparsity, cold-start users, ignoring users' specific spatial and temporal behaviors, not utilizing the shared behaviors of the users. In this paper, we propose a novel location recommendation method, namely Behavior-based Location Recommendation (BLR). BLR recommends a location to a user based on the users' repetitive behaviors and behaviors of similar users. Additionally, to better integrate the spatial information, BLR has two spatial components, a user-based spatial component to find the spatial preferences of the user, and a behavior-based spatial component to find locations of interest for different behaviors. Experimental studies on three real-world datasets show that BLR produces better location recommendations and can effectively address data sparsity and cold-start problems.

Keywords Social networks · Location-based services · Location recommendation · Collaborative filtering · Recommender systems

✉ Xin Wang
xcwang@ucalgary.ca

[1] Department of Geomatics Engineering, University of Calgary, 2500 University Dr. NW, Calgary, Alberta T2N 1N4, Canada

[2] Department of Electrical and Computer Engineering Engineering, University of Calgary, 2500 University Dr. NW, Calgary, Alberta T2N 1N4, Canada

1 Introduction

A Location-Based Social Network (LBSN) allows users to share their locations and location-related contents, such as notes and photos, with other users. A location recommendation service helps users explore new locations based on the preference models and spatial movement patterns extracted from the users' publicly shared information. On a LBSN, users' locational information is usually collected as check-ins which shows the location of the user at a certain time [1].

The growing popularity of LBSNs and abundance of shared location information has made location recommendation an active research area in the recent years — [1–15]. However, there are some deficiencies in the existing methods:

First, many of the current methods make recommendations to the users based on their own historical check-ins whereas some utilize the check-ins of the similar users. However, to find similar users, the current studies use the commonly visited locations and many of them do not consider the type of location or the time of check-in. In other words, current similarity measures lack temporal and spatial components. In addition, the collaborative model used in the literature cannot capture the global structures and relations of the user-location graph constructed from check-in data.

Second, some studies have treated locations as items in a regular recommendation system and can discover user-location relationships through general recommendation methods [6–9]. However, unlike other recommendation methods, in location recommendation, geographical constraints determine the reachability of a location to a user [16]. Therefore, the application of item recommendation methods with spatial constraints have been studied [10, 17–19]. These methods commonly utilize the same spatial probability distribution for all users in all regions. Because of this, they are not user-specific when it comes to modelling the spatial preference of the users. Human movements are also dependent on the behavior, users are willing to travel different distances for different activities. For example, a user might travel to the nearest store when shopping for groceries, whereas, she/he might prefer to travel farther to a fine restaurant. To summarize, spatial models should be personalized to model each individual user and based on the intended behaviours, which is missing in the previous location recommendation systems.

Finally, many existing location recommendation methods suffer from sparsity and the cold-start problems ([20, 21]). The enormous number of locations and users causes sparsity problem in location recommendation. Even the most active LBSN users can only visit a small portion of all locations and the most popular locations are only visited by a small number of users. Therefore, the user-location matrix used for building the recommendation model is usually very sparse. Cold-start problem happens when a user (cold-start user) or a location (cold-start location) is new to the LBSN. For a cold-start user, a small check-in history is recorded. Hence, the locational preferences of the user cannot be precisely modelled. A cold-start location is visited by a limited number of users, so the group of users interested in that location cannot be effectively detected.

In this paper, we approach the above-mentioned problems from a new perspective, namely the behavior-based perspective. In this perspective, we need to answer two questions before recommending a location. Q1) What does the user want to do at the given time? (What type of behaviour user likely to conduct?)Q2) How far does the user prefer to travel for the activity?

(How far will the user travel at this time? Where are the popular locations for the selected activities at this time?)

We propose a Behavior-based Location Recommendation (BLR) method that extracts users' behaviors from the check-ins and utilizes: (a) the repetitiveness of users' behaviours; (b) similarity of their behaviours; and (c) a novel spatial model which together result in a more effective location recommendation method. Specifically, our contributions are:

(1) To answer the first question, the concept of behavior is proposed, which is defined as the combination of a location type and time interval. Then we propose a Behavior Transition Model (BTM) to recommend locations by predicting the user's behavior using his/her behavior history and behaviors of similar users. To implement BTM, we create a user-behavior graph, then extend the Random Walk with Restart (RWR) method to find users who share common behaviors. BTM incrementally updates the user-behavior probability values based on past behaviors of the user and the similar behaving users. It considers the global structure of the user-behavior graph, thus utilizes all the information in the system when making the recommendation. BTM enables us to predict the user behavior even if he/she had not shown that behavior previously.
(2) Answering the second question requires knowing user's preferred areas at the given time and the popular location for each activity at that time. As the user travel range can vary based on the time and the activity the user is about to conduct, a new spatial component is proposed to utilize two aspects of human spatial behaviors, i.e. user reachability and location popularity. To model the user reachability, a user-specific spatial component generates a personalized temporal probability distribution for each user to quantify what areas are more interesting to the user at each given time. To model the location popularity, a behavior-based spatial model is proposed to find a spatio-temporal reachability model for each point of interest based on its local popularity. The more popular a location is, the greater distance users would like to travel to visit that location.
(3) In this paper, to better handle the sparsity and cold-start problem, we build a user-behavior matrix of a lower dimensionality rather than a user-location matrix in the recommendation. Using location type instead of the location, as long as the type of the cold start location is known, BLR can recommend the location based on the information from other locations of the same type. As for the cold start user, BTM can predict the user's behaviors based on the behaviors of other users. Also, the behavior-based spatial model leads him/her to popular locations based on his/her predicted behaviors.
(4) To evaluate the performance of the BLR, experimental studies have been conducted on three real-world datasets. We have discovered that the BLR method outperforms four existing location recommenders. In our experiments, BLR makes better recommendations with up to 30% higher precision compared with the current existing approaches. For cold-start users, BLR scores precision values two times higher than the baseline models. With the proposed spatial component, the BLR is shown to have reduced the spatial error to a third of the best existing location recommendation model.

The rest of this paper is organized as follows: Section 2 is literature review on recommendation and location recommendation methods. Section 3 introduces the BLR method. The experimental configuration and results are discussed in Section 4. Finally, the study is concluded in Section 5.

2 Related works

In the recent years, the location recommendation has been an active research area and different location recommendation methods have been proposed from different perspectives. Some of these models are based on learning the relationship of users and locations in a collaborative approach.

Zheng et al. [6] developed a method to mine interesting locations from the GPS traces of multiple users and find travel sequences for visiting those locations. They utilized a tree-based hierarchical graph to model individual check-in histories and applied a HITS (Hyperlink-Induced Topic Search)-based inference model to find the level of interest of a location and use it to find the interesting travel sequences. This method provides a good step-by-step approach to model locations' level of interest and finding the travel routes from the user-experience. However, this method lacks a spatial component to help generate a sequence that is suitable based on the current location of the user. Later, Zheng et al. [7] proposed the first study to connect locations and activities, and recommend locations based on the users' activities. They identified a user's activity by using user's comments, and then built a location-activity matrix and utilized user similarity into a user-centric CF model to make location recommendations. However, the method is difficult to be applied when user's comment is limited or not available.

Berjani and Strufe [8] proposed a regularized matrix factorization approach to make location recommendations to provide personalized location recommendations based on the entire check-in dataset. They used two inference strategies to interpret check-in data as user preferences. The first one used a simple binary preference definition, and the second one used the method of equal width intervals (EWI). This method provides a utilization of the check-in information and collaborative features in making recommendations. Zhou et al. [9] conducted a comprehensive study on different combinations of check-in data utilization and collaborative filtering algorithms on the location recommendation. They compared different check-in utilization methods (binary, probability, Frequency-Inverse Frequency) and a variety of Collaborative Filtering recommendation methods (user-based, item-based and PLSA) for the location recommendation. This was a comprehensive study of the application of different recommendation methods in location recommendation. However, both above methods lack spatial features to model the reachability of the locations.

Geographical constraints determine the reachability of a location. Recommending a location that is spatially unreachable to a user is a failed recommendation. Therefore, applying item recommendation methods with spatial constraints was the next trend in the related works. Ye et al. [10], Clemente and Bothorel [17] Lian et al. [18] and Yuan et al. [5] are examples of methods that include spatial features in their recommendation model. Ye et al. [10] proposed a method that accounts for user-location, user-user interactions as well as spatial reachability of locations. They model the spatial reachability as a power law probability distribution to select between candidate locations that generated from a user-based CF recommendation method. This method utilizes two types of user-user interactions to make recommendations and proposes a distance-based power law distribution for assessing the reachability of locations. However, this spatial probability is too general as it is the same for all users in all regions.

Clemente and Bothorel [17] created three graphs, namely the social graph, the frequentation graph and a geographic graph based on the check-in data and shopping center information. Next, they combined the three graphs to connect users and locations and used Katz centrality on the merged graph to compute a score for the candidate locations and recommended the top-N unvisited shopping centers. The method is an innovative approach to modelling spatial influence and utilizes social interactions in its recommendations. However, the method is limited to only one kind of user activity (e.g. shopping) and cannot adjust to other activities and other user preferences based on the time.

Lian et al. [18] utilized matrix factorization in a method called GeoMF. In this approach, the spatial features were added as constant values to the latent features matrix. The users' activity areas and locations' influence areas were determined and locations that are in the intersection of those areas were recommended to the user. Their approach of adding spatial information to the latent features is a novel approach and removes the need for a separate spatial model. However, with this approach a location that does not match the preferences of the user sometimes can still get a high rating because it is in the user's activity area. Additionally, this model can result in a very large and very sparse latent matrix that will decrease the efficiency of the recommender.

Yuan et al. [5] proposed a "co-pairwise ranking model" with the assumption that locations near previously rated ones are ranked higher by the user. The proposed method gave a positive rating to a location if it was visited by the user or the user had visited that location's neighborhood and a negative rating to other locations. Then they used a Bayesian Personalized Ranking method to model the user preferences and matrix factorization to find the underlying interactions between users and locations. Their method provided a good categorization of users' implicit feedbacks and it was well utilized in making location recommendations. In this method, the geographical neighborhood was used to model the spatial reachability, that is the users are likely to visit locations in the neighborhood of their previously visited locations. However, in this method, the neighborhood was defined as all locations in a pre-defined radius of the previously visited locations and the neighborhood threshold was the same for all locations. That is not usually the case as for example, the neighborhood of the user home should be larger than a distant restaurant the user visited once.

Adding the spatial features significantly improves the location recommendation methods. However, users' locational preferences are also impacted by the temporal factors. Therefore, many researchers added temporal aspects to their recommendation methods in addition to the spatial features to improve the location recommendation performance, such as Cho et al. [11], Rahimi and Wang [13], Gao et al. [14] and Rahimi et al. [1]. Cho et al. [11] proposed a method called Periodic Mobility Model (PMM) that considered both temporal and spatial components. In PMM, a user has a home location and a work location, and a user can be either in home or work state based on the time. PMM assumes that a user is more likely to visit locations near her home when she is in the home state and visit locations near her workplace when she is in work state. PMM provides a good combination of spatial and temporal factors for location recommendation. However, having only two states of home and work can limit the recommendation method as the user movements are very limited in the work state and can be much higher in the home state.

Rahimi and Wang [13] proposed the Probabilistic Category-based Location Recommendation method (PCLR) that used the user's repetitive temporal habits to predict the next category of location they would visit and made location recommendations accordingly. This method first predicted the location category the user would visit at the given time. It then recommended the most accessible ones to the user. This method provided recommendations to the user based on the prediction of the user behavior. However, it is prone to overfitting, as it can only predict location categories previously visited by the user. Also, it uses the same distance-based general spatial probability distribution for all users.

Gao et al. [14] used non-uniformness and consecutiveness properties to find a correlation between the check-in time and user preferences. They proposed the location recommendation framework with temporal effects (LRT) to utilize the temporal properties and user preferences to make location recommendations. This model first divided the user-location matrix into a set of sub-matrices and each sub-matrix only contained check-ins of one temporal state. Then, each sub-matrix was factorized into user preference and location characteristics matrices. After building the model, the results were aggregated to find the user preference for the given time and make recommendations. This method wisely uses temporal information to learn user's temporal preferences. However, temporal changes in location profiles was not modelled. Also, spatial reachability was not utilized in this method. More recently, Rahimi et al. [1] improved on their previous work by introducing the concept of behavior. Visiting a specific type of location at a specific time was defined as a behavior. They then utilized two collaborative filtering location behavior recommendation methods, namely Latent Behavior Analysis (LBA) and Behavior Factorization (BF), to predict the next behavior of the user based on his past behavior and the past behavior of similar users. These methods were then combined with a distance-based spatial component to provide location recommendations to the users. The two methods provide a good combination of user-specific and collaborative features in location recommendation. However, they utilize distance-based general spatial probability that is not temporal or user-specific.

Most Recently, Lian et al. [3] proposed an improved version of their joint geographical modelling and matrix factorization method for location recommendation. The new method, named GeoMF++, improved on the previous work by mapping the geospatial grids to low-dimensional latent space resulting in recovery of activity areas that were more meaningful and reasonable compared to GeoMF. Approaching the problem from a different angle, Geng et al. [2] proposed a two-step method utilizing multi-objective immune algorithm. The first step of this algorithm used social friends' information to improve collaborative filtering recommendation performance and generated candidate recommendations. The next step, location recommendation was modelled as a multi-objective optimization problem focusing on the similarity between check-in behaviors and geographical influences. A multi-objective immune algorithm was then used to optimize these two functions at the same time.

Overall, there are still holes to fill in location recommendation research. The user or behavior similarity are not well defined in the existing studies. Also, many studies have separated the temporal features from their spatial models. Because of that, unlike the spatial preferences of the users, the proposed spatial models do not change with time. That results in recommendation of locations that are not matching the user's preferences.

3 Methodology

In this section, we propose a new location recommendation method named Behavior-based Location Recommendation (BLR) that utilizes both spatial and temporal patterns of user behaviors.

3.1 Preliminaries and overview of the BLR

3.1.1 Preliminaries

Before further explanation of BLR, definitions of check-in and behaviour, as used in this research, are given.

Definition 1 A check-in is a tuple containing a user, a location and a timestamp.

$$c = (u, l, t) \tag{1}$$

Such a check-in shows that user u has visited location l at time t. In datasets, u is the unique user identifier, l is the unique location identifier and t is the time stamp in UTC.

Definition 2 A behavior b is a tuple consisting of the user's location category (cat) and the time interval (ti) of the time user was at the location, denoted as:

$$b = (cat, ti). \tag{2}$$

For example, if Amy checks into a Starbucks at 9:35 am, she has shown the behavior of (coffee shop, 9 am-10 am), which translates to "Amy visited a coffee shop between 9am and 10am".

Definition 2 enables us to find similar and dissimilar users as well as common behaviors among a group of users from their check-ins even if they are not visiting the same exact locations. If two users visit locations of the same category at similar times, they are considered as users with similar behaviors. For example, if Beth visits a Second Cup coffee shop at 9:15 am, she is showing a similar behavior to Amy. On the other hand, Carrie who only visits coffee shops around 4 pm is dissimilar to both Amy and Beth based on this behavior.

Definition 3 A Behavior Graph (BG) is an undirected weighted bipartite graph, denoted as:

$$BG = (B, U, E) \tag{3}$$

where B is the set of behavior nodes and U is the set of user nodes. E is the set of edges that connect users and their behaviors. Behavior graph is a weighted graph, and the weight of an edge represents the probability of the user showing the behavior.

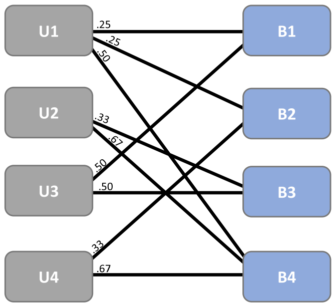
Behavior graph is a bipartite graph that represents the relationships between users and behaviors. The initial behavior graph is built using the check-in data and the weight of the edges are initialized to the normalized frequency of showing different behaviors. The sum of the weights of edges connected to each user is one.

Figure 1 shows an example initial behavior graph of four users and their behaviors based on the check-ins shown. For example, user U1's check-in history contains 4 check-ins. Two of these check-ins are at coffee shops but the time intervals of these check-ins are different. Thus, they are representing different behaviors (i.e. B1 and B2). The weight of the edge connecting U1 to B1 and B2 is 0.25 which is the number of times U1 has shown those behaviors (i.e. which is 1) divided by the number of check-ins by U1, that is 4.

3.1.2 A discussion on time interval

The behaviors are created as a combination of location category (cat) and time interval (ti). Location category is a constant value recorded in the dataset. In the following, we discuss different approaches to generating time interval.

User	Location	Time	Behavior
U1	Starbucks	8:28	B1 (Coffee Shop, 8am-9am)
U1	Second Cup	9:17	B2 (Coffee shop, 9am-10am)
U1	University	9:15	B4 (Office, 9am-10am)
U1	University	9:22	B4 (Office, 9am-10am)
U2	The Keg	13:33	B3 (Restaurant,1pm-2pm)
U2	Headquarters	9:22	B4 (Office, 9am-10am)
U2	Headquarters	9:05	B4 (Office, 9am-10am)
U3	Tim Hortons's	8:38	B1 (Coffee shop, 8am-9am)
U3	Ratatouille	13:14	B3 (Restaurant, 1pm-2pm)
U4	Municipality	9:11	B4 (Office, 9am-10am)
U4	Municipality	9:18	B4 (Office, 9am-10am)
U4	Starbucks	9:05	B2 (Coffee shop, 9am-10am)



(a) (b)

Fig. 1 **a** An example check-in dataset **b** the corresponding initial behavior graph

Time interval determination The length of the time interval is an important parameter for the behaviour generation. Therefore, in this section, we discuss two different approaches to determine the time intervals.

The equal-length approach is to use a constant value for time intervals, e.g. 30 min or 1 h. The equal-length approach results in a fixed number of time intervals with the same length, e.g. 48 or 24 time-intervals. This approach is simple and easy to implement. However, people are usually more active at certain time intervals compared to others. In other words, some time-intervals may have much more check-ins (e.g. rush hours or lunch and dinner time) compared to others (e.g. in the late night). To consider this characteristic, another approach called the equal-frequency method is proposed that generates time intervals with varying sizes so that each time interval can contain the same number of the check-ins in the original dataset. For example, we could generate 10 time-intervals and each interval has 10% of the total number of check-ins.

Check-in assignment to the time interval Two methods are developed for the time interval assignment, Boolean and fuzzy. In Boolean assignment, each check-in is only assigned to the time interval it falls into. For example, a check-in at 12:15 is assigned to the time interval 12:00-12:20. With Boolean assignment, a check-in is an event with a given time stamp. However, people tend to conduct the similar activities at the similar time. To reflect the temporal correlations among time intervals, the fuzzy assignment considers each check-in as a normal distribution in terms of time and the assignment depends on a membership value. In this study, the membership of the check-in to the time interval is the area under the curve of the membership-function bounded by the start and end points of the time interval. Based on the fuzzy assignment, a check-in at 12:15 can be assigned to time interval 12:00-12:20 with membership value of 75% and to the time interval 12:20-12:40 with membership value of 20%. With fuzzy assignment, a check-in can be viewed as an event spanning over several time intervals.

3.1.3 BLR framework

The behavior-based location recommendation method (BLR) framework is depicted in Fig. 2. BLR consists of two main components: a behavior prediction component and a spatial model component. Both components process the check-in data. The behavior prediction component uses the check-in history to build the Behavior Graph, which shows the probability of each behavior expected from users. The probability of conducting a behavior by the users is estimated by a Behavior Transition Model.

The spatial component, on the other hand, uses the check-in history to find Hot Check-in Areas (HCA) which are then used to learn behavior-based and user-specific spatial probabilities. When a user requests for recommendation, the recommendation system uses the behavior model, and two types of spatial probabilities, namely the user-specific and the behavior-specific spatial probabilities, to quantify the likelihood of visiting locations and recommends those with higher probability to the user. In the rest of this section we will discuss the BLR in details.

3.2 Behavior prediction component

The main task of behavior prediction component is to build a model to predict the behaviors of users at any given time. In this paper, the probability of the user showing a behavior at any given time is predicted using a collaborative filtering method. This method assumes that users will conduct behaviors previously shown by either themselves or user behaving similar to them.

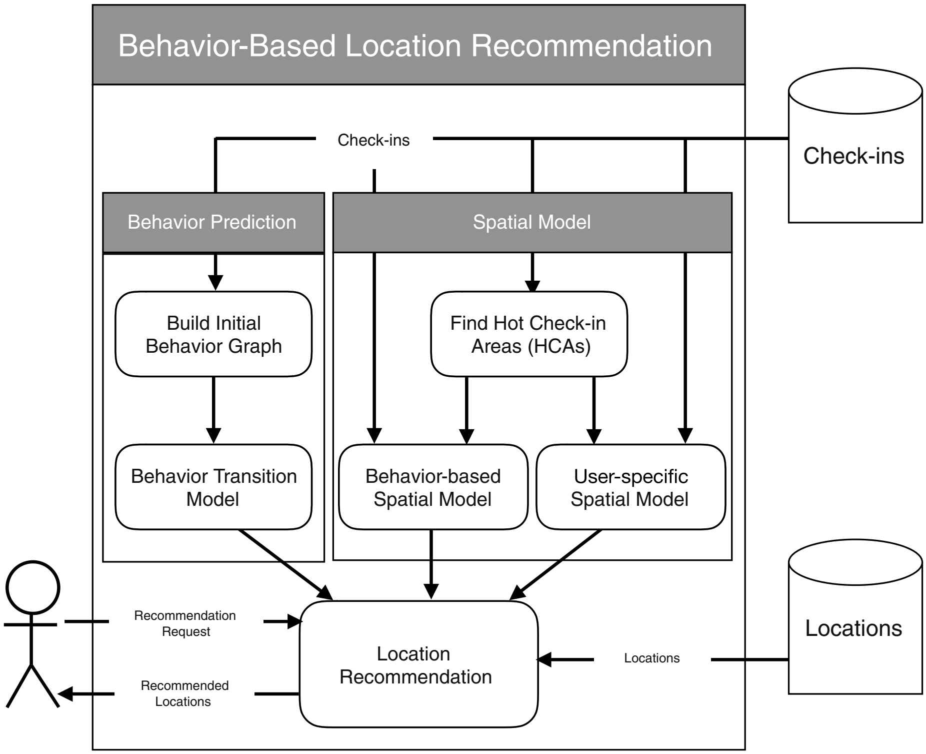


Fig. 2 Behavior-based Location Recommendation (BLR) framework

The first step of building behavior model is to build the initial behavior graph and initialize it using the check-in history. As mentioned, the initial behavior graph represents behavior of users extracted from their check-in history. It shows the historical behaviors of users and corresponding probabilities. For each user node, only some of the behavior nodes are connected to the user node.

The behavior transition model (BTM) is based on the Random Walk with Restart (RWR) method [4]. It utilizes the global structure of the initial behavior graph and estimates the probability values of all edges in the behavior graph by incrementally updating the user-behavior probability values based on past behaviors of the user and their similarly behaving users. In BTM, the behavior nodes connecting to a user node in the initial behavior graph is called the 1-step accessible nodes. The weight of the connecting edge is the probability of user conducting that behavior. Similarly, 2-step accessible nodes from each user node are nodes that can be accessed from the user node taking two steps. Which means that the 2-step accessible nodes of a user node u are user nodes that share at least one behavior with user u. This set also include the original user node u. The 2-step probability of two user nodes u_i and u_j showing the same behaviour can be calculated as the sum of the product of probabilities of u_i and u_j showing common behaviors, as shown in Eq. (4).

$$p_{2-step}\left(u_i, u_j\right) = \sum_{b_k \in B} p_{1-step}(u_i, b_k) p_{1-step}\left(u_j, b_k\right) \tag{4}$$

where $p_{1-step}(u, b_k)$ is the probability of user u showing behavior b_k, which is the weight of the edge connecting the corresponding user and behavior nodes on the initial behavior graph.

The 3-step accessible nodes of a user include the behavior nodes that the user is likely to do based on his/her own history and the behaviors of his/her 2-step similar users. The estimated 3-step probability of a user showing a behavior is defined in Eq. (5):

$$p_{3-step}(u_i, b_k) = \sum_{u_k \in U} p_{2-step}(u_i, u_k) p_{1-step}\left(u_j, b_k\right) \tag{5}$$

Equations (4) and (5) are similar recursive functions. So, we can derive the following general equations to calculate the probability between a user and behavior:

$$p_{2n-step}\left(u_i, u_j\right) = \sum_{b_k \in B} p_{(2n-1)-step}(u_i, b_k) p_{1-step}\left(u_j, b_k\right) \tag{6}$$

$$p_{(2n-1)-step}\left(u_i, b_j\right) = \sum_{u_k \in U} p_{2(n-1)-step}(u_i, u_k) p_{1-step}\left(u_k, b_j\right), \tag{7}$$

Similarly, the (2n-1)-step accessible behavior probability between a user and a behavior nodes can be calculated as the aggregate probability of user showing that behavior on the 1, 3, 5, or the $2n$-1 step. This way, the probability of a user showing a behavior is defined as:

$$p\left(u_i, b_j\right) = \lim_{n \to \infty} p_{(2n-1)-step}\left(u_i, b_j\right), \tag{8}$$

To calculate $p(u_i, b_j)$,all possible user and behavior combinations for the $p_{1-step}(u_i, b_j)$, $p_{2n-step}(u_i, u_j)$, $p_{(2n-1)-step}(u_i, b_j)$ and $p(u_i, b_j)$ can be represented using matrices. In BTM, the matrix containing all $p_{n-step}(u_i, n_j)$ is called the n-th step state matrix (Q_n) and Q_0 is the initial state matrix.

The initial state matrix shows the 0-step probabilities. Since the 0-step means no movement and BTM starts from user nodes, the entries corresponding to a user and itself have the value of 1 and the rest of entries is zero. The initial state matrix Q_0 is shown by Eq. (9).

$$
Q_0 = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_n \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{array}
\begin{array}{c}
\begin{array}{cccccccc} u_1 & u_2 & \cdots & u_n & b_1 & b_2 & \cdots & b_m \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0
\end{bmatrix}
\end{array} \tag{9}
$$

In BTM, the matrix containing all $p_{1-step}(u_i, b_j)$ is called the transition matrix (denoted by B). It can be built by converting the initial behavior graph into a matrix. Each row or column in the transition matrix represents a node in the behavior graph, so each matrix entry shows the probability of transition between the corresponding user and behavior nodes. Thus, the transition matrix B can be defined by Eq. (10).

$$
B = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_n \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{array}
\begin{array}{c}
\begin{array}{cccccccc} u_1 & u_2 & \cdots & u_n & b_1 & b_2 & \cdots & b_m \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & b_{u_1,b_1} & b_{u_1,b_2} & \cdots & b_{u_1,b_m} \\
0 & 0 & 0 & 0 & b_{u_2,b_1} & b_{u_2,b_2} & \cdots & b_{u_2,b_m} \\
0 & 0 & 0 & 0 & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & b_{u_n,b_1} & b_{u_n,b_2} & \cdots & b_{u_n,b_m} \\
b_{u_1,b_1} & b_{u_2,b_1} & \cdots & b_{u_n,b_1} & 0 & 0 & 0 & 0 \\
b_{u_1,b_2} & b_{u_2,b_2} & \cdots & b_{u_n,b_2} & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & 0 & 0 & 0 & 0 \\
b_{u_1,b_m} & b_{u_2,b_m} & \cdots & b_{u_n,b_m} & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array} \tag{10}
$$

The entry b_{u_i,b_j} in the behavior transition matrix B indicates the probability of user u_i showing the behavior b_j. Since the edge weights on the initial behavior graph is normalized, we know that $\sum_{j=1}^{m} b_{u_i,b_j} = 1, 1 \le i \le n$.

By multiplying the transition matrix and a current-step state matrix, the next-step state matrix is formed in BTM. The corresponding entry of user u and behavior b in state matrix shows the calculated probability of user u showing behavior b at the current step of the state matrix. Behavior graph is a bipartite graph, we have $b_{u_i,u_j} = 0, 1 \le i \le n$ and $b_{b_i,b_j} = 0, 1 \le i \le m$.

Since starting from a user node, the random walker can be on a behavior node only at odd steps, in BTM we are only interested in odd-step probabilities. Thus, we can infer the matrix update equations given in Eqs. (11) and (12).

$$
Q_1 = BQ_0 \tag{11}
$$

$$
Q_{2n-1} = (1-p)BBQ_{2(n-1)-1} + pQ_1, (n \ge 2) \tag{12}
$$

where Q_{2n-1} is the matrix showing the (2n-1)-step probabilities and p is the restarting probability which helps in adjusting the probability values and adding more weight to the behaviors that are accessible with a lower number of steps.

The BTM training algorithm iteratively updates the state matrix using the above update equations until the matrix converges (i.e. the entries of the matrix change less than a given convergence threshold θ). In the other words, the final state matrix Q can be calculated by Eq. (13),

$$Q = \lim_{n \to \infty} Q_{2n-1} \tag{13}$$

We can estimate Q with the Q_θ defined in Eq. (14).

$$Q_\theta = Q_{2n-1} \text{ where } |Q_{2n-1} - Q_{2(n-1)-1}| < \theta \tag{14}$$

The pseudocode for BTM is given in Fig. 3. The input of BTM algorithm includes the initial behavior graph, the restarting probability and the convergence threshold. The first line builds the transition matrix from the initial behavior graph. Then the initial state and the first-step state matrices are built in lines 2-3. The loop on the lines 6 to 10, iteratively updates the state matrix until it does not change more than the pre-specified threshold. Finally, the behavior graph is built using the final state of the random walker and is returned. The behavior graph can then be used to assign the behavior probability values for each user. Also, to improve memory and time complexity of BTM, Optimized Random Walk with Restart (ORWR) [22] was used.

3.3 Spatial model

The second step of BLR is to build a spatial model based on the predicted behavior. The spatial model describes the probability of a user visiting a location at the given time based on the user preference and the reachability of the location. More specifically, the spatial model of BLR is responsible for finding locations matching the predicted user's behavior and within user's proximity.

In this section, a new spatial model is proposed, which consists of two sub-models, a user-specific spatial model and a behavior-based spatial model. Both sub-models are represented as the two-dimensional spatial probability distributions and their combination shows the probability distribution of locations based on a user's own behavior and the influence of other users.

3.3.1 Detecting hot check-in areas

Since each local area has a unique check-in distribution, and people's preferences differ from one geographic region to another, the user-based and behavior-based spatial models should be built for each geographic region. Thus, Hot Check-in Areas (HCA) are identified, HCAs represent areas with high density of check-in destinations.

Algorithm 1 BTM

1: $B \leftarrow iBG$ ▷ iBG is the initial behavior graph
2: $Q_0 \leftarrow queryMatrix(B)$ ▷ Create a diagonal unity matrix with the same dimensions as B
3: $Q_1 \leftarrow BQ_0$
4: $i \leftarrow 1$
5: $\epsilon \leftarrow \infty$ ▷ ϵ is the estimation error
6: **while** $\epsilon > \theta$ **do** ▷ θ is the convergence threshold
7: $\quad i \leftarrow i + 1$
8: $\quad Q^i \leftarrow (1-p)BBQ^{i-1} + pQ_0$ ▷ p calculate the next step state matrix
9: $\quad \epsilon \leftarrow ||Q_i - Q_{i-1}||_F$
10: **end while**
11: **return** MakeBehaiorGraph(Q_i)

Fig. 3 Pseudocode for the Behavior Transition Model

To identify HCAs, all visited locations are first extracted from the check-in data set of the LBSN. Then the locations are clustered into HCAs using a spatial clustering method. Since HCAs do not have a pre-determined shape and should be determined by the density of check-ins, a density-based spatial clustering algorithm DBSCAN is selected to perform the clustering in this study [23]. The method first finds the best values for two parameters ϵ and *minPts* using the k-dist method [23]. ϵ specifies how close the points should be to be considered neighbors and minPts specifies the minimum number of neighbors a point should have to be considered in a cluster. The parameter values are then utilized by the DBSCAN to find the extent of the Hot Check-in Areas. After identifying HCAs, the next step of building the spatial model is to build the user-specific and behavior-based spatial components based for each HCA.

3.3.2 User-specific spatial probability

Since users have personal spatial preferences, the spatial model requires a user-specific spatial model to make proper personal location recommendations. To build the user-specific spatial model for each HCA, all user check-ins in an HCA are selected. Then a kernel density estimation function is built by averaging the values of kernel functions centered on each check-in location in the HCA. Formally, the user-specific spatial probability $p_{s_u}(x, u|h, t)$ of user u visiting location x given the HCA h and time t is defined as:

$$p_{s_u}(x, u|h, t) = \sum_{c \in C_{u,h,t}} \frac{1}{|C_{u,h,t}|} k_u(|x - c.l|) \tag{15}$$

where $|C_{u, h, t}|$ is the number of user u's check-ins in the HCA h in a time interval covering t. The $c.l$ represents location of check-in c. $k_u(x - c.l)$ represents the value of k at location x given $c.l$ is the center of the kernel function.

The user-specific kernel function k_u is defined as:

$$k_u(x) = \frac{m_u e^{-p_u x} + m_G e^{-p_G x}}{2} \tag{16}$$

where $m_u e^{-p_u x}$ is the power law estimation of user's check-ins and $m_G e^{-p_G x}$ is a global power law estimation representing all check-ins in the dataset, it is used to avoid overfitting if a user has limited check-ins.

Figure 4a shows the check-in locations of a user (shown as stars) on a university campus between 9 and 10 am along with the number of check-ins to each of those locations. Given these check-ins, using Eqs. (15) and (16) we can build a user-specific probability distribution for the user shown in Fig. 4b. The user is more likely to visit locations that are marked with higher probability values, in this case, locations near L1 have higher user-based probability values.

3.3.3 Behavior-based spatial probability

Behavior prediction component predicts the probability showing a behavior by an individual user. It uses the check-in history of the users for a given behavior to measure how likely another user is to visit a location of the same behavior. Depending on the behavior, users might travel longer or shorter distances. To learn the behavior-based spatial probability, all check-ins matching the behavior are first selected. Using a two-dimensional kernel probability distribution, the probability of visiting each location in the HCA is then determined. Formally, given a set of check-ins in HCA h matching behavior b the behavior-based spatial probability of visiting location x which matches behavior b, $p_{s_b}(x|h,b)$ is defined as:

$$p_{s_b}(x|h,b) = \sum_{c \in C_{b,h}} \frac{1}{|C_{b,h}|} \; k_b(|x - \mathrm{c}.l|) \tag{17}$$

where $|C_{\mathrm{b},\,h}|$ is the number of check-ins in HCA h that correspond to behavior b and $c.l$ represents the location of check-in c. $k_b(|x - \mathrm{c}.\ l|)$ represents the value of k_b at location x given $c.l$ is the center of the kernel function.

Similar to the user-specific kernel function, the behavior-based kernel function k_b is defined as:

$$k_b(x) = \frac{m_b e^{-p_b x} + m_G e^{-p_G x}}{2} \tag{18}$$

where $m_b e^{-p_b x}$ is the power law estimation of check-ins matching behavior b and $m_G e^{-p_G x}$ is a global power law estimation representing all check-ins in the dataset, it is used to avoid overfitting if a limited number of check-ins match the behavior.

For example, Fig. 5a shows the locations checked in matching the behavior of (coffee shop, 9 am-10 am) in a synthetic dataset created to simulate students' check-ins on the main campus of the University of Calgary. Stars show the check-in locations, and the numbers in brackets show number of check-ins into those locations. Among all locations matching the selected behavior, L2 is the most popular location for the behavior which had been visited 853 times, whereas that L1 and L3 were visited 314 and 226 times, respectively. Figure 5b shows the contour map of the behavior-based spatial probability distribution derived using Eqs. (17) and (18). Since L1 and L2 are close to each other, the behavior-based spatial probability of showing the selected behavior is higher in their vicinity. The neighbourhood of the location L2 is the most probable area with a probability value close to 0.60.

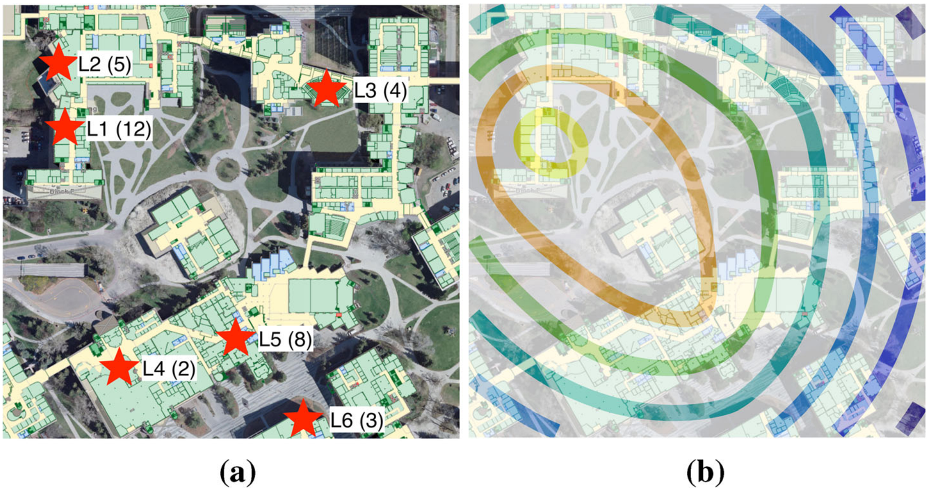


Fig. 4 **a** A sample user's check-ins between 9 and 10 am (shown as stars) **b** user-specific spatial probability distribution for that user between 9 and 10 am

Figure 6 shows the pseudocode for building the spatial model using a set of check-ins. The first two lines find the Hot Check-in areas. Lines 7 to 10 find the parameters required to calculate the user specific probabilities for each HCA in each time interval. Lines 11 to 15 find the parameters to calculate the behavior based spatial probabilities for each HCA and behavior. Finally, line 18 returns the spatial model consisting of user-specific and behavior-bases spatial models.

3.3.4 Calculation of the spatial probability

The user-specific spatial probability determines the locations the user is likely to visit in each HCA based on his/her own past behaviors. The behavior-based spatial probability takes account of all users' behaviors, which helps recommend locations outside the user's past behavior areas, thus helping them explore new locations. In addition, the behavior-based spatial probability helps recommend locations to the cold-start users. Combining the user-specific spatial probability and behavior-based spatial probability, the spatial probability is able to predict the nearby locations matching the predicted behavior of the user.

The user-specific spatial probability and behavior-based spatial probability generated by the kernel density estimation functions have the continuous values for the whole HCA area, but only the points of interest (POIs) are recommended to the user. Therefore, given the predicted behavior of the user, the spatial probability of a user is defined.

Definition 4 Given time t and a behavior b, the spatial probability p_s of user u visiting location l is defined as the product of user-specific spatial probability and behavior-based spatial probability:

$$p_s(u, l|h, t) = \begin{cases} p_{s_u}(l, u|h, t) \cdot p_{s_b}(l|h, (l.type, t)) & , l \text{ is a POI} \\ 0 & , l \text{ is not a POI} \end{cases} \tag{19}$$

where p_{s_b} and p_{s_u} are the behavior-based and user-specific spatial probabilities, respectively. If the location is not a POI, the spatial probability will be set to zero. In this way, POIs that are

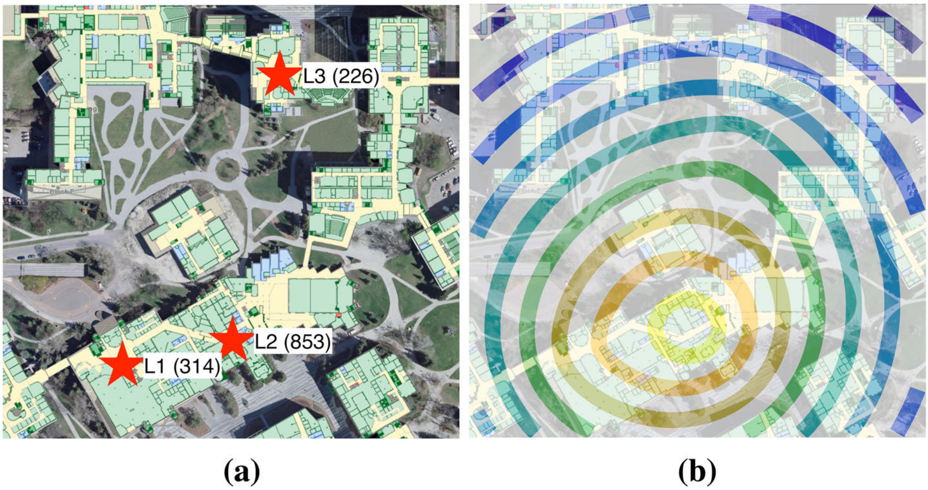


Fig. 5 **a** Locations matching the behavior of (coffee shop, 9 am-10 am) shown in red star **b** behavior-based spatial probability distribution for that behavior

Algorithm 2 Spatial Model

```
Determine ϵ and minPts                          ▷ Use k-dist method (Ester et al. 1996)
HCAs ← DBScan(c, ϵ, minPts)                     ▷ c is the set of all check-ins
USS ← new UserSpecificSpatialModel
BBS ← new BehaviorBasedSpatialModel
for h in HCAs do
    for ti in TimeIntervals do
        for u in Users do
            c ← check-ins by u to locations in h at ti
            set parameters of USS_{u,h,ti} using check-ins in c
        end for
        for cat in LocationCategories do
            b ← behavior(cat, ti)
            c ← check-ins to locations in h matching behavior b
            set parameters of BBS_{b,h} using check-ins in c
        end for
    end for
end for
return SpatialModel(HCAs, USS, BBS)
```

Fig. 6 Pseudocode for building the spatial model

more reachable to the user and are more likely to be visited for the predicted behavior of the user will have the higher spatial probability.

For the example area shown in Figs. 4b and 5b with the user-specific and behavior-based spatial probability distributions, respectively, the spatial probability of the whole area is shown in Fig. 7a and b shows the candidate POIs for the user for the behavior of (coffeeshop, 9 am-10 am) with the corresponding spatial probability values.

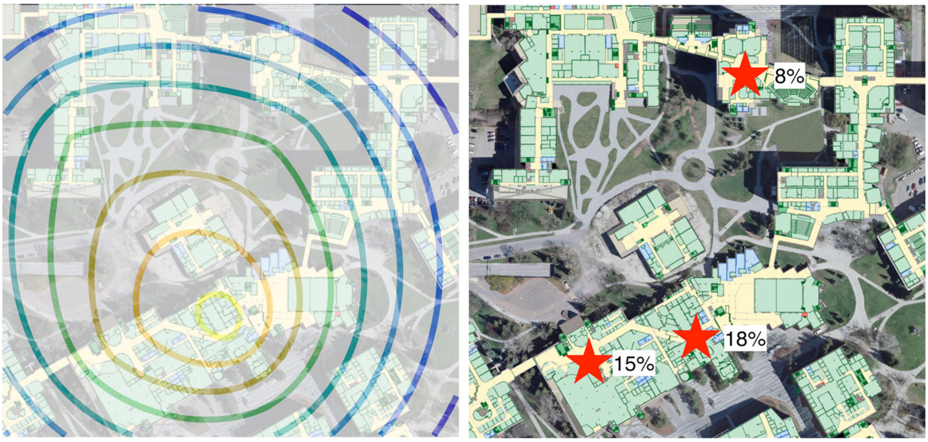
3.4 Recommending locations to the user

To recommend a location using BLR, the behavior and spatial probabilities of visiting the location by the given user are calculated as discussed above. The expected check-in probability is defined by Definition 5. Locations with overall higher expected probability are recommended to the user.

Definition 5 The expected probability of a user u checking in to the location l at the given time t is defined as:

$$p(u, l|t) = p_b(u, (l.type, t)) \cdot p_s(u, l|h(l), t) \tag{20}$$

where $p(u, l| t)$ is the probability of the user u checking into location l at time t. $p_b(u, (l.\ type, t))$ is the probability of the user u showing the behavior made using the category of location l and time interval of t. $p_s(u, l| h(l), t)$ is the spatial probability of user u visiting location l given the HCA location l belongs to and the time t.



(a) (b)

Fig. 7 **a** Contour map showing the spatial probability distribution of the sample user showing behavior of (coffee shop, 9 am-10 am) **b** Candidate recommendation locations for the sample user for the behavior of (coffee shop, 9 am-10 am) with corresponding spatial probability values

4 Experimentation and evaluation

In this section, the proposed BLR method is evaluated on three real-world location-based social network datasets and the results are compared with six recent location recommendation models. All recommendation methods are implemented in MATLAB and the experiments were conducted on a 2.5 GHz Core i7 Macbook Pro having 16GB of RAM and a 480GB SSD.

4.1 Data description

Three real-world check-in datasets, namely Gowalla [9], Foursquare [24, 25, 26] and Brightkite [11, 12], are used for the experiments. Details of these datasets are given in Table 1.

As shown in the Table 1, the Gowalla dataset contains 5462 users, 5999 locations and 104,851 check-ins. The other two datasets Foursquare and Brightkite respectively contain 266,909 and 51,406 users, 3,689,126 and 772,966 locations and 33,278,683 and 4,747,281 check-ins.

As for the categorical information, Gowalla dataset includes two levels of location categories with 10 high level categories such as Food, Shopping and about 300 low level categories such as coffee shop, restaurant and fast food. The original Foursquare and Brightkite check-in datasets do not include the categorical information, so the Google maps reverse geocoding service (Geocoding API [24]) were first used to map the original check in latitude and longitude to the street address. Then Google search API [25] were applied to search for the

Table 1 Dataset Details

Dataset	Number of Users	Number of Locations	Number of Check-ins	Location Category Information
Gowalla	5462	5999	104,851	Yes
Foursquare	266,909	3,680,126	33,278,683	No
Brightkite	51,406	772,966	4,747,281	No

address and find the category of the business at the street address. After the above data processing, locations in both Foursqure and Brightkite datasets are of 107 categories in total.

For the experiments, each dataset is first separated into a training dataset and a testing dataset. The testing dataset contains one randomly chosen check-in of each user. The training dataset, on the other hand, contains all the remaining check-ins. To remove the effect of random selection, five different testing and training dataset pairs are generated from each dataset. All experiments are performed on all five pairs of datasets and the average values are reported.

To compare the proposed BLR with the existing location recommendation methods, six of the most recent location recommendation methods are implemented, including Periodic Mobility Model (PMM) proposed by Cho et al. [11, 12]; User-based CF, Social influence, Geographical influence (USG) model proposed by Ye et al. [10]; GeoMF++ [3], GeoMF [18], USPB [29], and MLR [2].

PMM utilizes a user-specific two-state spatial component. It has a temporal component to determine the state of the user and a spatial component to determine the location of the user based on the state. USG uses user similarity, influence of friends and geographical influence to make location recommendations. GeoMF++ learns users' preferences by combining user preferences and spatial preferences into one matrix and using Matrix Factorization to learn a model. GeoMF takes the same approach as GeoMF++ to keep a constant part in the user and location latent features for the spatial interactions, whereas in GeoMF++ it is fused into the location latent features. USPB, on the other hand, utilizes social influence, proximity and naïve Bayesian classification for making recommendations.

MLR, uses a two-step method to learn a collaborative filtering model and then uses a multi-objective immune algorithm to find the best balance between user similarity and geographical influences.

4.2 Time interval study

One of the main components of this study is the time interval. The behaviors are created as a combination of location type and time interval. The location type is a constant value given by the dataset. The starting point and the length of time interval, however, should be provided before the learning of the model begins. We compare the performance of different time interval strategies and choose the best performing one for the rest of the experiments. To compare different strategies, we used the precision of the recommendation defined as:

$$Precison = \frac{|Recommended\ Locations \cap Correct\ Locations\ |}{|\ Recommended\ Locations\ |} \tag{21}$$

It measures the ratio of the recommended locations that are visited by the user.

4.2.1 Time interval start and length

In this experiment, we compare the impact of the two approaches i.e., equal-length and equal-frequency, in terms of the recommendation performance. One-hour is used as time interval for the equal-length method. For the equal-frequency method, 25 time-intervals are generated for all three datasets, and each of the time interval contains 4% of the total check-ins. The lengths of time intervals range from 20 min to 5 h.

Figure 8 shows the comparison result on the precision of the recommendation for the two different approaches. As shown, the equal-frequency time interval approach results in precision improvements of at least 9% higher than the equal-length time interval. The reason of the improvement is that the equal-frequency method determines the length of the time intervals based on the check-in distribution. With the equal-frequency time intervals, the recommender system can better differentiate behaviors for the time intervals with more intense activities compared with the equal-length method.

4.2.2 Check-in membership to the time interval

As mentioned, the assignment of check-ins to time intervals can be either Boolean or fuzzy. In this experiment, we compare the recommendation performance between the Boolean and fuzzy check-in assignments. As for the fuzzy assignment, a normal distribution membership function is used with the mean value of the check-in time. The membership of the check-in to the time interval is the area under the membership-function, curve given the start and end points of the time interval. For such a fuzzy membership function, we first conduct a sensitivity analysis and compare the results using different time intervals, and then select 7.5 min as the standard deviation for the time interval membership function because it shows the best performance.

Figure 9 compares the precision of the recommendation for Boolean and fuzzy assignment of time intervals. As shown in the figure, the fuzzy assignment results in at least 5% improvement over the Boolean assignment. The reason that fuzzy assignment outperforms the Boolean assignment is that the fuzzy assignment better utilizes the temporal correlations. More specifically, these two features help us better model the user-behavior interactions for short time intervals.

4.3 Comparison of the location recommendation models

4.3.1 Location recommendation performance

In this experiment, we measure the quality of the recommendations of the proposed BLR and compare with six baseline methods, i.e., USG, PMM, GeoMF, USPB, GeoMF++ and MLR.

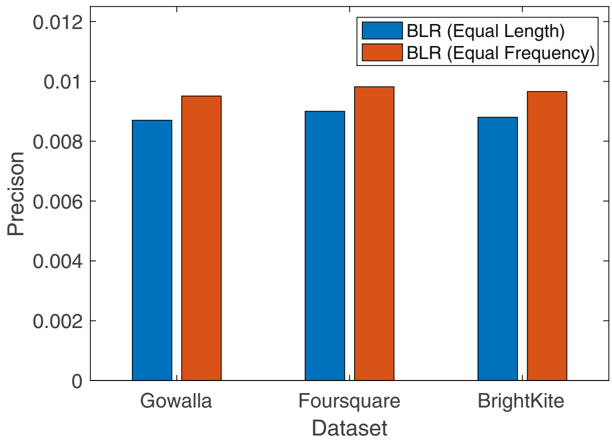


Fig. 8 Comparison of the performance of BLR using equal-length and equal-frequency time intervals

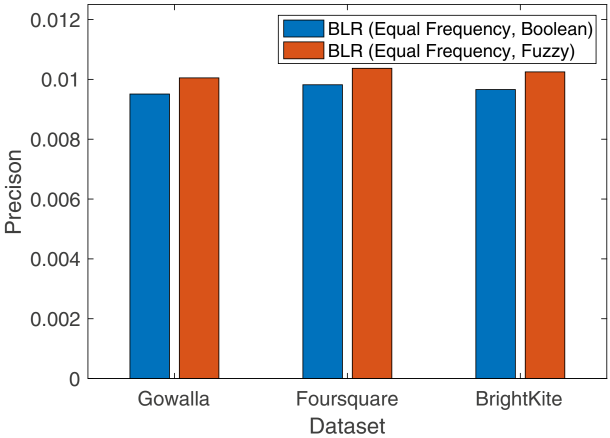


Fig. 9 Comparison of Precision with Boolean and Fuzzy Assignment of Time Intervals

To quantify and compare the quality of the recommendations we use precision and recall. Precision is previously defined in Eq. (21), recall is defined Eq. (22).

$$Recall = \frac{|Recommended\ Locations \cap Correct\ Locations\ |}{|\ Correct\ Locations\ |} \tag{22}$$

Precision measures the portion of the recommended locations that match the actual checked-in location. Recall, on the other hand, is the portion of the visited locations predicted by the location recommendation model.

Figure 10 depicts the precision values for each model recommending N locations. The results show that BLR outperforms the six baseline methods in terms of precision. As shown in Fig. 10, the precision of the BLR ranges from .0094 to .0176, which is about 13% higher than the best performing baseline method GeoMF++ with the precision values ranging from .0078 to .0014. Comparing precision values over different values of N, we can see that the relative performance over different datasets is similar. The only difference visible is that the precision decreases as the number of recommendations (N) increases. That is because despite scoring more correct recommendations with N increasing, that increment is not enough to offset the increase in the number of recommendations. Thus, the overall recommendation precision still decreases.

The reason the BLR performs the best is that it considers the global structure of the user-behavior graph for making behavior recommendations and its recommendations are not only based on the immediate similar users but all users and behaviors on the check-in dataset. The spatial model in BLR also predicts the target location based on user's history and the popularity of the locations based on the predicted behavior.

GeoMF++ is the best performing baseline method. GeoMF++ and GeoMF are not performing well as they do not consider the temporal preferences of the users. MLR utilizes a user-location matrix as well as a location-location matrix to make recommendations. This means parameters like distance and reachability which affect the users' decision are not used for recommendation in MLR. In USPB the spatial component is only based on the distance and does not consider the effect of user's activity on the distance he/she is willing to travel. The main issue of PMM is not using user similarity as well as being limited to home and work states for the users. They used the classic definition of similar users in recommendation system

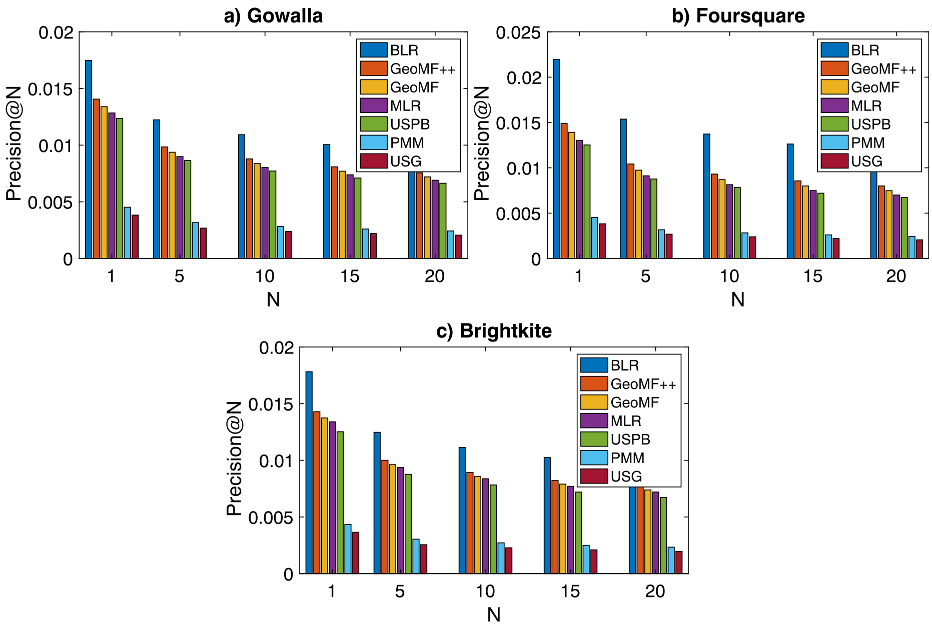


Fig. 10 Precision values of the location recommendation models recommending N locations. **a** Gowalla, **b** Foursquare **c** Brightkite

which is not adapted to the requirements of the location recommendation. Also, they utilized a general power-law distribution to model the geographical influence which is not user or behavior specific.

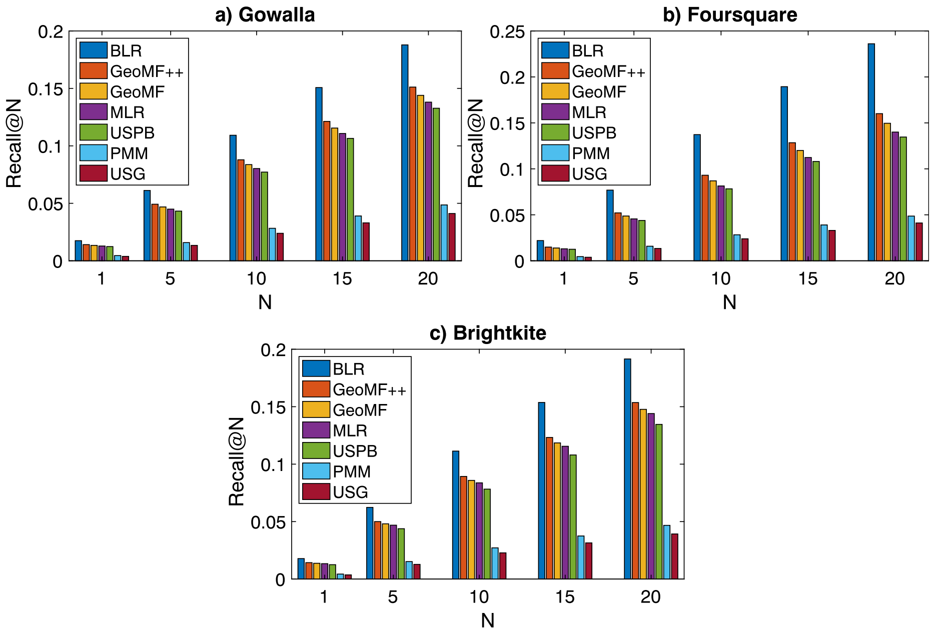


Fig. 11 Recall values of the location recommendation models recommending 15 locations. **a** Gowalla, **b** Foursquare **c** Brightkite

Figure 11 shows that recall values follow similar trends. Except that the recall value is increasing with increasing N. That is because increasing the number of recommendations help make more correct recommendations, however, the number of correct locations is not affected by this change hence the recall value improves.

4.3.2 Behavior prediction performance

As stated earlier, BLR consists of two main components: a behavior prediction component and a spatial model component. The behavior prediction component predicts the behavior of users at any given time. In this experiment, we evaluate how well the proposed BLR behavior prediction component performs. Behavior is the combination of location category at the given time. So, for the baseline models we use the category and time of the recommended location to find the predicted behavior.

To measure the performance of the behavior prediction, the precision of a behavior recommendation is defined by Eq. (23):

$$Behavior\ Precision = \frac{|Predicted\ Behaviors \cap Correct\ Behaviors\ |}{|\ Predicted\ Behaviors\ |} \tag{23}$$

The behavior prediction precision is the ratio of the correctly predicted behaviors to all predicted behaviors.

Figure 12 shows the precision values of the recommendation methods recommending N behaviors. As shown in the Fig. 12, the precision of the behavior prediction component of BLR ranges from 0.08 to 0.148. In comparison, the best performing baseline model

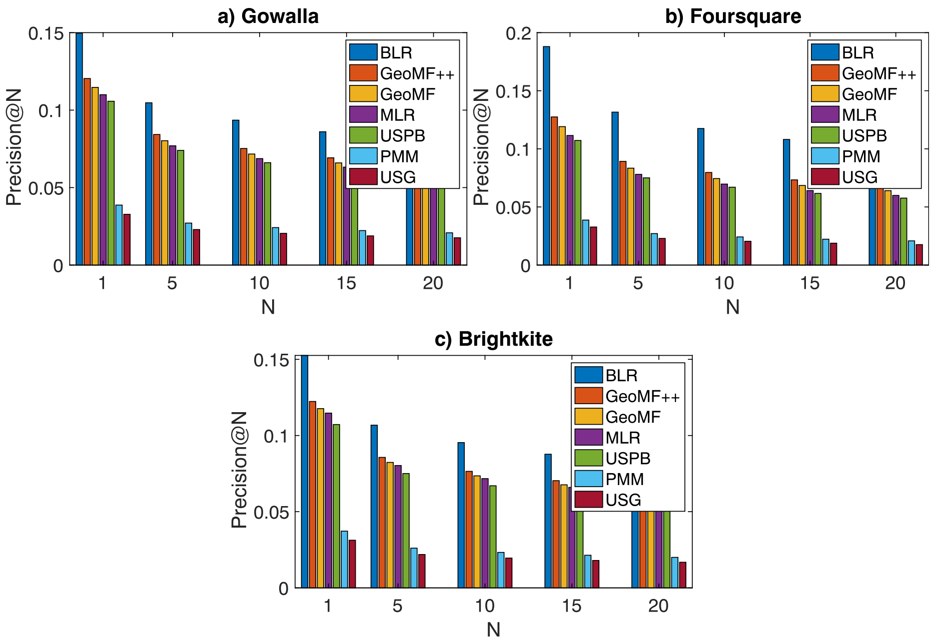


Fig. 12 Precision values of recommending behaviors based on behaviors recommended with top 15 locations. **a** Gowalla **b** Foursquare **c** Brightkite

(GeoMF++) has precision values ranging from 0.062 to 0.122 and the worst performing one (USG) has precision values in the range of 0.018 and 0.032.

Similar to the location recommendation performance, the behavior prediction component of BLR outperforms all baseline models in terms of precision of the behavior prediction. Another observation is comparing Figs. 10 and 12, the precision value is higher for recommending behaviors than recommending locations. That is, in many cases the correct behavior is predicted, but the final recommended location is not the one chosen by the user. For example, the behavior prediction correctly predicts that the user will be visiting a fast food and recommends fast food X, but the user actually visits the neighboring fast food Y. This behavior is expected because the number of candidate locations is much higher than the number of behaviors. Hence it is more likely for a recommendation method to have a false location recommendation than to have a false behavior recommendation.

4.3.3 Spatial performance

In BLR, the spatial component finds a location matching the predicted user behavior. From the previous experiment, the behavior prediction has the higher precision than the location recommendation. That is, the models predicted the user behavior correctly but failed to predict the visited location. The objective of this experiment is to evaluate the spatial components of the proposed BLR and compare with other six baseline location recommendation methods.

Given a set of recommended locations, the spatial precision of recommendation is defined as:

$$Spatial\ Precision = \frac{1}{1 + \frac{\sum_{c \in D} dist\left(\hat{l}_c, l_c\right)}{|D|}} \tag{24}$$

where D is the set of check-ins in the dataset and |D| is the number of the check-ins in D. c is a check-in selected from D. $\hat{l}_c$ is the recommended location for the user of check-in c at the timestamp of that check-in and l_c is the corresponding location of check-in c.

Spatial precision is the inverse of the average distance of the recommended location and visited locations. That means the higher the spatial precision, the closer the predicted location

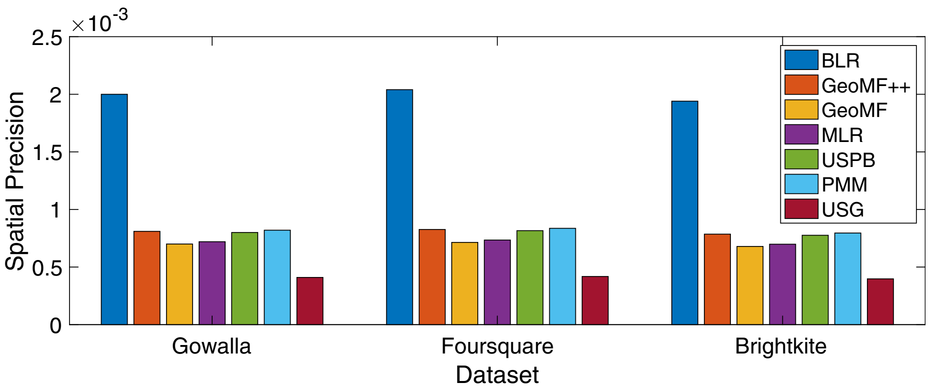


Fig. 13 Comparison of the spatial proximity values of the location recommendation methods. **a** Gowalla **b** Foursquare **c** Brightkite

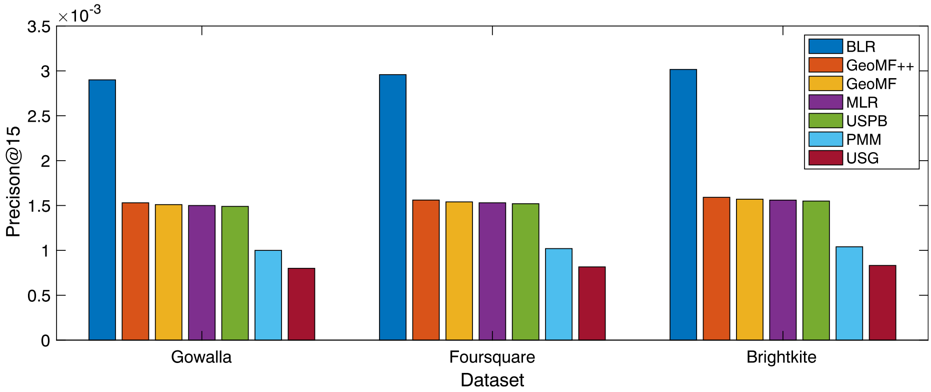


Fig. 14 Precision of location recommendation models recommending 15 locations to cold-start users. **a** Gowalla **b** Foursquare **c** Brightkite

is to the visited location. Figure 13 shows the spatial precision values of these location recommendation methods.

As shown in Fig. 13, the spatial precision of the BLR ranges between 0.0019 and 0.0021. The best baseline method is PMM with the spatial precision ranging from 0.00080 to 0.00083 whereas the worst performing baseline USG has spatial precision values ranging from 0.00040 to 0.00042. BLR outperforms the baseline models by a margin of 150% (for PMM) and 500% (for USG). Among all location recommendation methods, BLR utilizes a more complex spatial component to find behavior-based and user-specific spatial probability models to find a suitable candidate for recommendation. The spatial components of USG, GeoMF, GeoMF++, USPB and MLR do not consider the changes in user movement pattern based on time and type of location they are visiting, PMM improves them by considering the state of the user (Home, Work), but modelling users' intentions into just two states of home and work is not sufficient to make a general model.

4.4 Performance of the location recommendation model on cold-start users

One of the main obstacles in recommendation systems is the cold-start problem. One main objective of this study is to provide a better recommendation model for the cold-start users. In this experiment, performance of location recommendation models is compared for cold-start users. To measure the recommendation performance on cold-start users we use the location recommendation precision defined in 4.2.1.

We define cold-start users as users with less than five check-ins in the dataset. To measure the cold-start performance, we build the model using all users in the training dataset but measure the precision value only for the recommendations made to the cold-start users. The precision value is reported in Fig. 14. Compared precision values in Fig. 14 to the precision values for the all users shown in Fig. 10, the precision values are lower for the cold-start users. However, the precision of the BLR for cold start users ranges from 0.0029 to 0.0030 and the best baseline recommendation method GeoMF++ has the precision value ranging between 0.00151 to 0.00163. We can observe that BLR still outperforms all baseline models and has widened the gap with the best performing baseline model GeoMF++ to 90% (from 13%). This is because BTM uses the information from the more active users and unseen behaviors of the cold-start users can be derived through the similarities they have with the active users.

To find the significance of the improvement in precision, we use the t-test analysis over the 5 observations of precision values for all recommendation models for the whole dataset as well as the cold-start user dataset. The results suggest that the precision values of the proposed BLR model results are significantly higher than the precision values of the baseline location recommendation models given $\alpha = 0.05$. This indicates that the proposed model is statistic significantly outperforming the baseline models.

5 Conclusions an future works

In this paper, a new location recommendation method, Behavior-based Location Recommendation (BLR), is proposed. BLR predicts the behavior of the user based on their past behaviors as well as the behaviors shown by similar users. BLR then recommends a location suitable for the predicted behavior based on the user-specific and behavior-specific spatial models. Both spatial models are based on Kernel Density Estimation and utilize historical data to find and recommend reachable locations. The user-specific model helps the location recommendation model predict location of user and locations in their proximity. The behavior-based model helps location recommendation model find hot-locations for the behaviors the user might show. This new spatial probability is shown to reduce the spatial error three times more than the best performing baseline method.

BLR outperforms the baseline methods of PMM, USG, GeoMF, GeoMF++, MLR and USPB on check-ins from three real-world datasets. We have shown that BLR results in precision values that are 25% to 300% higher than the baseline methods. This shows that, although people have different spatial behaviors, they have common temporal behaviors. Finding the common temporal behaviors and adding specific spatial constraints, we can model the spatial behaviors with higher accuracy.

The proposed models are not dataset specific and can be applied to any check-in dataset with category information. For the Gowalla dataset, category information is already collected and incorporated into the dataset. If such information is not recorded in the check-in dataset, as it is for Brightkite and Foursquare datasets, it can be collected from other location services such as Foursquare, Yelp or Google Maps.

The context of the time interval such as workdays/weekends, seasons, weather, also play a role in the behavior analysis. For example, time interval 8 am-9 am for a workday is different from the same time interval on a weekend or holiday. Similarly, the same time interval in the summer time might be different in the winter as users could engage in different activities. A context-aware approach can be used to differentiate the same time interval on a rainy day from a clear weather. Such contextual information complicates the model but can potentially increase the performance of the model. Therefore, studying the contextual variables, and modelling them effectively is a future plan.

An interesting extension of the current BLR method is to incorporate a 3D spatial model. Such model can effectively recommend locations in different levels of a building, but it requires having elevation information in the check-in dataset. Current model recommends locations to the users based on the every-day activities. This recommendation model is effective for most cases. However, when users travel to a new location, then using day-to-day behavior models for making recommendations may not be effective. For example, during daytime, Parisians go to their offices, however, recommending an office in Paris to a user who lives in New York and is visiting Paris will not be a good recommendation. The location recommendation model should be able to detect that this user is travelling and does not need to go to work. Improving the current model to effectively recommend locations for travelling users is a next step in this study. Additionally, we will continue to compare the proposed BLR method with the more recent and complex location recommendation methods.

Acknowledgements The research is supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant to Xin Wang and Behrouz Far, and National Natural Science Foundation of China (No. 61772420).

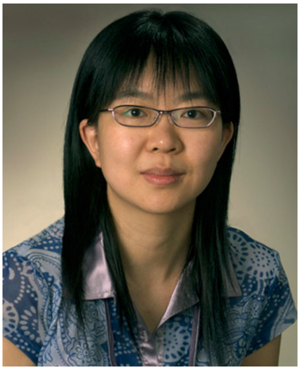
References

1. Rahimi, S.M., Wang, X., and Far, B. (2017) Behavior-based Location Recommendation on Location-Based Social Networks, The 21st Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2017), Jeju Island, Korea, May 23-26, 2017
2. Geng B, Jiao L, Gong M, Li L, Wu Y (2019) A two-step Personalized Location Recommendation based on Multi-objective Immune Algorithm. Inf Sci 475:161–181
3. Lian D, Zheng K, Ge Y, Cao L, Chen E, Xie X (2018) GeoMF++: Scalable Location Recommendation via Joint Geographical Modeling and Matrix Factorization. ACM Trans Inf Syst 36(3):33:1–33:29
4. Bagci H, Karagoz P (2016) Context-aware location recommendation by using a random walk-based approach. Knowl Inf Syst 47(2):241–260
5. Yuan F, Guo G, Jose J, Chen L, Yu H, Chen L (2016) Joint geo-spatial preference and pairwise ranking for point-of-interest recommendation. In: Proceeding of: 28th International Conference on Tools with Artificial Intelligence (ICTAI 2016), San Jose
6. Zheng Y, Zhang L, Xie X, Ma WY (2009) Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th international conference on World Wide Web (pp. 791-800). ACM
7. Zheng VW, Zheng Y, Xie X, Yang Q (2010) Collaborative location and activity recommendations with GPS history data. In: Proceedings of the 19th international conference on World Wide Web (pp. 1029-1038). ACM
8. Berjani B, Strufe T (2011) A recommendation system for spots in location-based online social networks. In: Proceedings of the 4th Workshop on Social Network Systems. ACM
9. Zhou D, Wang B, Rahimi SM, Wang X (2012) A study of recommending locations on location-based social network by collaborative filtering. In: Advances in Artificial Intelligence (pp. 255-266). Springer Berlin Heidelberg
10. Ye M, Yin P, Lee WC, Lee DL (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (pp. 325-334). ACM
11. Cho E, Myers S, Leskovec J (2011a) Friendship and Mobility: User Movement In Location Based Social Networks. In: 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1082-1090. San Diego
12. Cho E, Myers S, Leskovec J (2011b) Data from: Friendship and Mobility: User Movement in Location Based Social Networks [dataset] available form: https://snap.stanford.edu/data/loc-brightkite.html [Accessed 19 June 2018]
13. Rahimi SM, Wang X (2013) Location Recommendation Based on Periodicity of Human Activities and Location Categories. In: Advances in Knowledge Discovery and Data Mining (pp. 377-389). Springer Berlin Heidelberg
14. Gao H, Tang J, Hu X, Liu H (2013) Exploring temporal effects for location recommendation on location-based social networks. In: Proceedings of the 7th ACM conference on Recommender systems, 93100. ACM
15. Bao J, Zheng Y, Mokbel MF (2015) Recommendations in location-based social networks: a survey. Geoinformatica 19(3):525–565
16. Yu C, Liu Y, Yao D, Ding Q (2015) Mining User Check-in features for Location Classification in Location-based Social Networks. IEEE Symposium on Computers and Communication (ISCC), Larnaca, pp. 385-390
17. Clemente RP, Bothorel C (2013) Recommendation of shopping places based on social and geographical influences. In: RSWeb 2013: 5th ACM RecSys Workshop on Recommender Systems and the Social Web
18. Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y (2014) GeoMF: joint geographical modelling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14). ACM, New York, 831-840

19. Yuan Q, Cong G, Sun A (2014) Graph-based point-of-interest recommendation with geographical and temporal influences. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (pp. 659-668)
20. Park MH, Hong JH, Cho SB (2007) Location-based recommendation system using Bayesian user's preference model in mobile devices. In: Ubiquitous Intelligence and Computing (pp. 1130-1139). Springer Berlin Heidelberg
21. Jesús Bobadilla, Fernando Ortega, Antonio Hernando, Jesús Bernal, (2012) A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based Systems 26:225-238
22. Rahimi SM, Silva R, Far B, Wang X (2019) ORWR: Optimized random walk with restart for recommendation systems. Proceeding of the 32nd Canadian Conference on Artificial Intelligence, Kingston
23. Ester, M., Kriegel, H-P., Sander, J., Xu, X. (1996) A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, August 02-04, 1996, Portland, Oregon
24. Yang, D., Zhang, D., Qu, B., (2015a) Participatory Cultural Mapping Based on Collective Behavior Data in Location Based Social Networks. ACM Trans. on Intelligent Systems and Technology (TIST)
25. Yang D, Zhang D, Chen L, Qu B (2015b) NationTelescope: Monitoring and Visualizing Large-Scale Collective Behavior in LBSNs. Journal of Network and Computer Applications (JNCA) 55:170–180
26. Yang D. Global Check-in dataset. Available at: https://sites.google.com/site/yangdingqi/home/foursquare-dataset [Accessed 19 June 2018]
27. GeoCoding API by Google Maps, https://developers.google.com/maps/documentation/geocoding/start [accessed 19 June 2018]
28. Search API. Google API. https://developers.google.com/custom-search/json-api/v1/overview [Accessed 19 June 2018]
29. Kunhui L, Jingjin W, Zhongnan Z, Yating C, Zhentuan X (2015) Adaptive location recommendation algorithm based on location-based social networks. In: Proceedings of International Conference on Computer Science and Education, pp. 137–142 (2015)

Seyyed Mohammadreza Rahimi is a PhD candidate at the Department of Geomatics Engineering, University of Calgary. He started the PhD program after he received a master's degree from the same department in 2014. He has a BSc in Software Engineering and has been working on spatial data mining especially the location recommendation systems since 2011.

Behrouz H. Far has been a Professor of software engineering with the Schulich School of Engineering, University of Calgary, since 2001. His research interest areas are engineering of intelligent, distributed and heterogeneous networked systems, implementation, testing and verification of agent-based software systems. He established Intelligent Software Systems Laboratory, Schulich School of Engineering, University of Calgary.



Xin Wang is an Associate Professor at the Department of Geomatics Engineering, University of Calgary. She holds a B.Sc. in Computer Science, M.Eng. in Software Engineering from Northwest University, China and a Ph. D in Computer Science from University of Regina, Canada. Her current research interests are spatial data mining, ontology and knowledge engineering in GIS, web GIS and location-based social networks.