Hindawi
Security and Communication Networks
Volume 2017, Article ID 9127612, 11 pages
https://doi.org/10.1155/2017/9127612

Research Article

Reliable Collaborative Filtering on Spatio-Temporal Privacy Data

Zhen Liu,[1,2] Huanyu Meng,[1] Shuang Ren,[1,2] and Feng Liu[1,2]

[1]*School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China*
[2]*Engineering Research Center of Network Management Technology for High Speed Railway of MOE, Beijing 100044, China*

Correspondence should be addressed to Huanyu Meng; huanyum@bjtu.edu.cn

Received 8 October 2017; Revised 24 November 2017; Accepted 6 December 2017; Published 28 December 2017

Academic Editor: Zhiping Cai

Lots of multilayer information, such as the spatio-temporal privacy check-in data, is accumulated in the location-based social network (LBSN). When using the collaborative filtering algorithm for LBSN location recommendation, one of the core issues is how to improve recommendation performance by combining the traditional algorithm with the multilayer information. The existing approaches of collaborative filtering use only the sparse user-item rating matrix. It entails high computational complexity and inaccurate results. A novel collaborative filtering-based location recommendation algorithm called LGP-CF, which takes spatio-temporal privacy information into account, is proposed in this paper. By mining the users check-in behavior pattern, the dataset is segmented semantically to reduce the data size that needs to be computed. Then the clustering algorithm is used to obtain and narrow the set of similar users. User-location bipartite graph is modeled using the filtered similar user set. Then LGP-CF can quickly locate the location and trajectory of users through message propagation and aggregation over the graph. Through calculating users similarity by spatio-temporal privacy data on the graph, we can finally calculate the rating of recommendable locations. Experiments results on the physical clusters indicate that compared with the existing algorithms, the proposed LGP-CF algorithm can make recommendations more accurately.

1. Introduction

With proliferation of mobile phones and location-based services (LBS), the existing social networks are able to collect the users geographical position in real time. LBS is combined with the traditional social network to form the location-based social network (LBSN). Through use of location services, LBSN integrates the online virtual network with the offline real world, thereby enabling users to share and obtain information of their interest more easily. Due to this reason, LBSN is gaining more and more favor with users. In addition to the relationship information of the traditional social network and the self-labelled information of the user, LBSN encompasses the user-registered historical trajectory collected via GPS and the labelled information of relevant locations [1]. Furthermore, the users mobile behavior patterns and trajectory show some characteristics in terms of time frequency, geographical distance, social relationship, and content [2–5]. For example, friends are more likely to check in together at the same place. The locations where the user checks in on a daily and weekly basis also show some characteristics.

LBSN makes it possible for location recommendation. Location or Point of Interest (POI) refers to a geographical point that is useful or interesting to the user, such as hotel, restaurant, museum, and supermarket. Location recommendation refers to the service where the user-location check-in record is used to predict the location that has never been visited by the user but might be of supreme interest to the user and then recommend this location to the user [6].

Examples of LBSN include Foursquare, Gowalla, and Google Latitude [7]. This type of services allows the user to publish information on the place where they are via check-in and share their experience. These services have attracted a myriad of users. Statistical data indicates that Foursquare has more than 55 million users by the end of 2014. About 600 thousands to 1 million users check in at Foursquare each day. Over 6 billion check-in data items have been collected [6], which involve multidimensions, including the geographical

location (Geo) automatically collected by the users mobile devices, temporal data, and content data [8].

The increasing number of LBSN users is accompanied by the sharp rise in the amount of multidimensional LBSN check-in data, making it more difficult for users to filter the information. In this context, how to recommend custom location more accurately by combining the abundant spatio-temporal information of the LBSN check-in data and the behavior tracks with the traditional recommendation algorithms is an issue of great significance. Meanwhile, due to increase in the size of check-in data, the recommendation algorithm imposes higher demands on the backend computational ability. The traditional single-machine computation method and the open-source Hadoop-based computation method are more and more computationally inefficient and resource intensive [1]. How to provide the user with real-time location information by developing a new recommendation algorithm that can process big data efficiently is a fresh challenge to the social recommendation system [9]. After the advent of the distributed parallel graph framework proposed in recent years, the graph theories have been used by the academia and industry to model the relationship between social network data. Moreover, based on the efficient graph framework, the graph algorithm is used to support machine learning and data mining, enabling the problem to be solved much more quickly.

Based on our previous work [10], we propose a new collaborative filtering-based location recommendation algorithm for LBSN, LGP-CF, which is based on parallel graph calculation and clustering, using the spatio-temporal data of LBSN. Taking the users check-in behavior patterns into account, the proposed method segment the dataset and obtains the set of users similar to the target user using the clustering algorithm to reduce the range of choices for similar users. User-location bipartite graph is established via the check-in data. The users common trajectories in the graph are filtered based on propagation of message across the graph. Afterwards, the trajectory data of the set of similar users that can represent the spatio-temporal information is combined with the point data to compute the similarity between the target user and each of similar users. Finally, the locations are clustered using the longitude and latitude data. The shortest path algorithm is then used to determine the set of recommended locations quickly and reliably. The final rating of a location is computed using the temporal information regarding the visit of similar users to it.

The contribution of this paper can be summarized as follows:

(1) The graph theory is used to model the spatio-temporal information and the user trajectory information of LBSN, facilitating rapid location of users and check-in locations in the graph.

(2) Calculation of user and location similarity is optimized after combining with the spatio-temporal privacy information of LBSN and taking the point and trajectory data into account.

(3) In addition to the point and trajectory of data, we also consider the regional data to cluster users and locations and reduce the size of data that needs to be computed.

The rest of this paper is organized as follows. Section 2 discusses related work. In Section 3, LBSN data analysis is presented; then data representation and modeling are given. Based on the data model, this paper proposes users similarity calculation using point, trajectory, and regional data information in Section 4. Section 5 is the proposed algorithm and its parallel design and implementation, followed by the experiment and evaluation results in Section 6. Conclusion is finally given in Section 7.

2. Related Works

Generally, user check-in location involved in LBSN recommendation can be classified into the following categories [11].

(1) Point Data. It is the most common type of user check-in location. It is characterized by fixed longitude and latitude of the geographical location corresponding to the point data. This type of data is usually used to compute the physical distance between users and measure the interuser similarity. Many point data-based recommendation methods have been proposed for LBSN [12–14].

(2) Trajectory Data. With proliferation of smart phones capable of localization, the mobile trajectory that the user follows during a time period can be recorded and then used by the academia and industry to study the users continuous behavior. Many trajectory data-based algorithms have been proposed to recommend routes for navigation and tour [15–17].

(3) Regional Data. The geographical locations can be divided into different regions according to predefined criterion. Data analysis and feature extraction are performed on each of the regions to facilitate user recommendation. Alternatively, the clustering methods can be used to cluster the collected data and produce regional data before recommendation.

The above analysis of user mobility behavior performed using various types of data indicates that user mobility is usually constrained by geographical space and social relationship. Lian et al. [2] and Liu et al. [4] reported that the user is more inclined to move within a geographical space nearby rather than go to a distant place during a time period. The results obtained by mining and analyzing the Brightkite and Gowalla check-in data indicate that 20% of the continuous check-in behaviors happen within a radius of 1 km, 60% happen within a radius of 1 km to 10 km, and only 20% happen beyond a range of 10 km. Zhang and Wang [18] revealed that the geographical location and time of check-in behavior are very periodic for most users. According to the work by Lian et al. in [2], the range of user mobility is centered on two points, that is, home and work place, which means that the user seldom moves to a place beyond a range of the two centers. In [19], Aamir jointly considered user mobility trajectory, regional data, social popularity, and custom location recommendation. A tree-based layered classification model based on the trajectory data was established. The regional data was clustered in each layer. The popularity of a location during a time period was computed using the ratio of users who have checked in at the location to all users of the same class.

Table 1: Example of check-in data of Gowalla dataset.

User ID	Check-in time	Altitude	Longitude	Location ID
0	2010-10-19T23:55:27Z	30.2359091167	−97.7951395833	22847
0	2010-10-18T22:17:43Z	30.2691029532	−97.7493953705	420315
3353	2010-10-04T06:12:33Z	39.7478004013	−104.9992454052	1109125
4368	2010-02-21T03:11:51Z	37.7625977833	−122.4231266667	174904
29534	2010-04-01T03:42:08Z	31.10072965	−97.44364675	821666
80157	2010-05-31T13:51:11Z	48.199R39617	16.3874741445	57426

Table 2: Example of check-in data of Foursquare dataset.

User ID	Check-in time	Altitude	Longitude	Location ID
0	2011-01-01	41.727575	−88.031988	0
1	2011-01-01	51.31791	−0.588761	1
2	2011-01-01	33.767021	−84.352638	2
3	2011-01-01	40.774759	−73.982432	3
4	2011-01-01	40.77476	−73.9824	4
5	2011-01-01	26.93896	−82.0532	5

Most of the graph model-based algorithms are reliant on clustering. After assuming that there is correlation between users with similar preferences, the graph clustering algorithm classifies the nodes according to node property and correlation [20]. In [12], Yao et al. proposed a collaborative location recommendation framework CLR for LBSN. In their framework, GPS is first used to obtain user trajectory data in a three-layer (user-location-behavior) structure. Afterwards, a graph model is established that consists of three types of nodes (user, location, and behavior). Finally, the proposed algorithms are used for collaborative filtering and location recommendation. In [21], Jin et al. proposed a model based on link analysis and custom PageRank. The user in the dataset is regarded as the node in the directed graph; mutual following between users is regarded as the edge. The custom PageRank algorithm is used to compute the rank of recommendable locations for the target user during a time period. In [22], Cui et al. proposed a new location recommendation algorithm based on the graph model. In their method, the vertex consists of user vertex and location vertex, while the edge encompasses the friendly relationship between users and the user-location check-in relationship in the historical information. The user vertexes that have made friendship with the target user are sorted out according to their similarity with the target user. Afterwards, location is recommended by sorting out the location vertexes which have been visited by these friends but not visited by the target user.

3. Representation and Modeling of Check-In Data

3.1. Check-In Dataset Analysis. Gowalla is a LBSN website where users check in to share their current locations with friends. It consists of 196,591 nodes and 950,327 edges. The Gowalla dataset used in the experiment includes 6,442,890 records collected from February 2009 to October 2010. Examples of Gowalla user check-in data are given in Table 1.

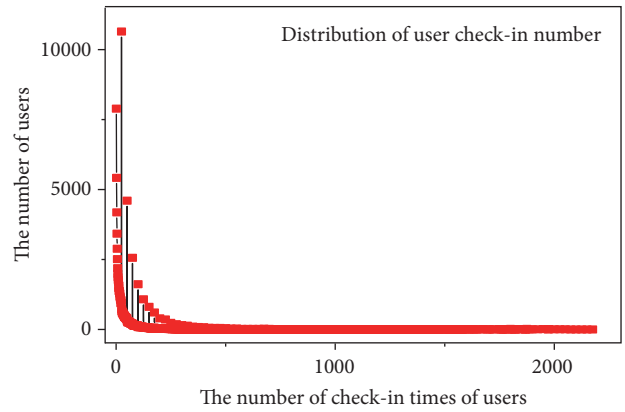

Figure 1: The distribution curve of user check-in numbers.

Foursquare is a LBSN service which encourages mobile phone users to check in and share their current locations with friends. It effectively combines the traditional social network with mobile Internet. Its dataset includes user check-in data, user social data, and user residence data. Foursquare includes 1,385,223 pieces of user check-in data. Examples of Foursquare user check-in data are given in Table 2.

The total number of checked-in users is 107,092 in the Gowalla dataset. The number of user check-in times is 60 in average, 2,175 at most, and 1 at least and mostly falls within the range [1, 300]. The number of corresponding users decreases with increase in the check-in times within this range. The distribution of all user check-in times is shown in Figure 1.

The number of Gowalla user check-in locations is also computed. It is indicated that there are 1,280,969 checked-in locations in Gowalla. The number of location check-in times is 5 in average, 5,811 at most, and 1 at least. The distribution of all location check-in times is shown in Figure 2. From this figure, it can be seen that most of the locations are seldom

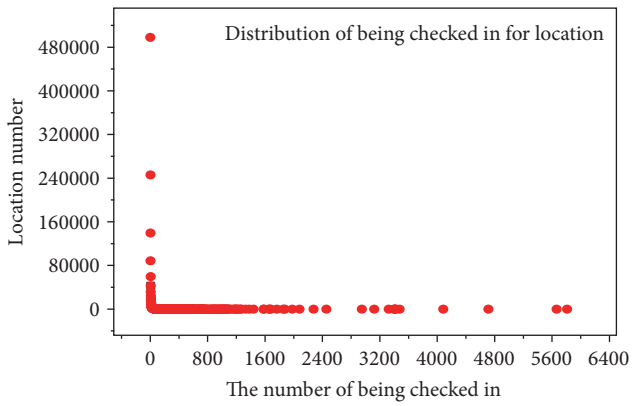

Figure 2: The distribution curve of the checked location numbers.

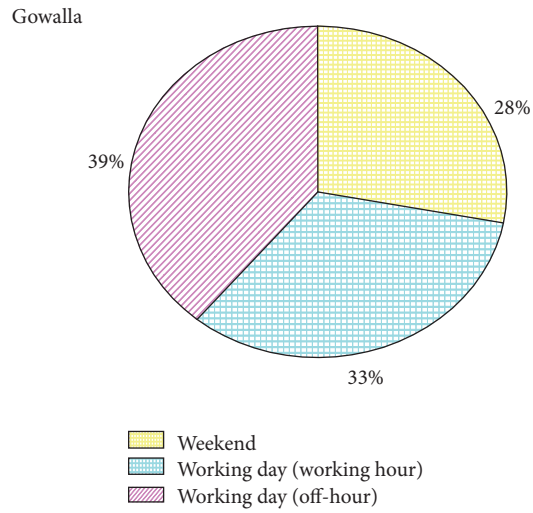

Figure 3: The distribution curve of user check-in number in a different period.

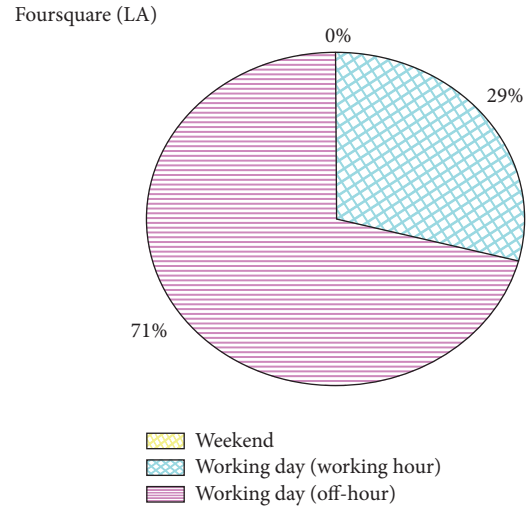

Figure 4: The distribution curve of user check-in number in a different period.

checked in. This means that the user check-in data is very sparse.

Performance of the recommendation system is closely related to data sparsity. Recommendation accuracy will be seriously affected if the data is very sparse. Therefore, we focus on the recommendation results of methods rather than the absolute value of performance metrics.

3.2. Check-In Data Representation. Let u, p, and t denote the user, location, and check-in time, respectively. Also let $U = \{u_1, u_2, \ldots, u_n\}$ denote the set of users in the user-location check-in data, $P = \{p_1, p_2, \ldots, p_m\}$ denote the set of locations, and $T = \{t_1, t_2, \ldots, t_l\}$ denote the set of check-in time. Hence, each of the location check-in data can be represented with a 5-dimension vector $\overrightarrow{d_{ijk}} = [u_i, p_j, p\mathrm{Lng}_j, p\mathrm{Lat}_j, t_k]$, where $i \in [1 \cdots n]$, $j \in [1 \cdots m]$, $k \in [1 \cdots l]$, t_k denotes the check-in time of the user u_i at the location p_j, $p\mathrm{Lng}_j$ denotes the longitude of the location p_j, and $p\mathrm{lat}_j$ denotes the latitude of the location p_j. If the user u_i has never visited the location p_j, the vector $\overrightarrow{d_{ijk}}$ does not exist. All vectors form the set of user-location check-in data, D. The recommendation algorithm is responsible for predicting the possibility that the user visits the location that he has never visited before using the dataset D.

3.3. Segmentation of the User Check-In Dataset. It can be learned that the geographical location and time of check-in behavior is very periodic for most users [18]. The range of user mobility is usually centered on two points, that is, home and work place, which means that the user seldom moves to a place beyond a range of the two centers [2]. Considering user needs for recommendation at different time periods, experiments are performed on the Gowalla and Foursquare datasets. Figures 3 and 4 show the ratio of the number of user check-in times at different time periods to the total number of check-in times in the Gowalla and Foursquare datasets, respectively.

In this paper, the check-in data is divided according to time periods. Demographic statistics analysis indicates that the location visited by the user during working days is very different from the location visited during the weekend. Moreover, the checked in location by the user during the working hours differs greatly from that checked in after hours. Therefore, the dataset D is divided into three segments in this paper.

Weekend dataset DoffDay: it refers to the dataset whose check-in time is the weekend in the user check-in dataset, and $D\mathrm{offDay} = \{\overrightarrow{d_{ijk}} \mid t_k \in T\mathrm{offDay}\}$, where ToffDay denotes the time set of weekend. Working hour dataset DofficeTime: it refers to the dataset whose check-in time is the working hour in the user check-in dataset, and $D\mathrm{officeTime} = \{\overrightarrow{d_{ijk}} \mid t_k \in T\mathrm{officeTime}\}$, where TofficeTime denotes the time set of working hours. The time period from 8:00 a.m. to 7:00 p.m. is defined as the working hour in the demography. Off-hour dataset DNofficeTime: it refers to the dataset whose check-in time is the off-hour of working day in the user check-in

dataset, and $DN\text{officeTime} = \{\overrightarrow{d_{ijk}} \mid t_k \in TN\text{officeTime}\}$, where TNofficeTime denotes the set of off-hour during working day. The time period apart from working hours during the working day is defined as the off-hour in the demography.

The weekend dataset DoffDay, working hour dataset DofficeTime, and off-hour dataset DNofficeTime are all the subsets of the user check-in dataset D: that is, $D = D\text{offDay} \cup D\text{officeTime} \cup DN\text{officeTime}$. Moreover, the intersection set of any two of the three subsets is empty. Considering the request time period by the target user, we filter a subset D' from D: that is, $D' = D\text{offDay}$ or $D' = D\text{officeTime}$ or $D' = DN\text{officeTime}$. Recommendation is made to the target user by performing data mining and analysis of D'.

3.4. Users Clustering on Temporal Pattern and Spatial Region. Each user either has some or no similar preferences to the target user. Based on this observation, the selected check-in data subset is clustered into two classes according to the target user. The class of the target user consists of similar users.

There is correlation in two users who share similar preferences [23]. Based on this assumption, property vector is constructed for each member of the user set in the selected check-in data subset. And these users are clustered according to the property vector. The property vector involves the number of check-in times, time pattern, and spatial region. It can be written as

$$\overrightarrow{v_i} = \langle u_i, l\text{MaxLng}, l\text{MaxLat}, l\text{NearLng}, L\text{NearLat}, l\text{DistLng}, L\text{DistLat}, l\text{MaxWeek}, l\text{MaxTime}\rangle, \quad (1)$$

where u_i denotes the ID of current user, lMaxLng and lMaxLat denote the longitude and latitude of the location lMax most frequently checked in by the user u_i, lMaxWeek denotes the day of a week that the user u_i most often checks in at the location lMax, and lMaxTime denotes the hour of a day that the user u_i most often checks in at the location lMax. Describing user behavior in detail is helpful in identifying similar users more accurately. In addition to spatio-temporal description of the users mostly frequently registered location, we draw inspiration from the results in [2, 4], taking user mobility range into account. Let lNearLng and lNearLat denote the longitude and latitude of the location that the user has once checked in and has the shortest Euclidean distance to lMax. Also, let lDistLng and lDistLat denote the longitude and latitude of the location that the user has once checked in and has the longest Euclidean distance to lMax. After construction of the user property vector, the k-means algorithm is used to extract users from the class of the target user as the similar user set Sim$U$.

3.5. Graph Modeling for User-Location Data. After extracting the similar user set Sim$U$ through clustering, we need to filter the user check-in data subset D' that is selected according to the recommendation request time. If u_i corresponding to the element $\overrightarrow{d_{ijk}}$ of D' does not belong to the similar user set Sim$U$, the element should be deleted from D'. The filtered user check-in data subset is called D'_1. In this paper, we model the filtered subset D'_1 as a user-item (location) bipartite graph. The graph G is shown in Figure 5. While clustering the similar users according to the regional data, we jointly consider the impact of point data and trajectory data on recommendation when we construct the graph model.

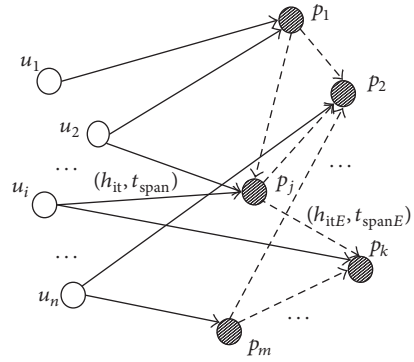

Figure 5: User-location bipartite graph.

The user check-in data subset D'_1 is represented with the graph G, and $G = \langle V, E\rangle$, where V denotes the set of vertexes, including user vertexes and location vertexes. $V = U_g \cup P_g$, where $U_g = \{u_1, u_2, \ldots, u_n\}$ denotes the user set in D'_1 and $P_g = \{p_1, p_2, \ldots, p_m\}$ denotes the location set in D'_1. E denotes the set of edges, including the edge between the user and the registered location and the edge between locations. $E = E_v \cup E_t$, $\langle u_i, p_j\rangle \in E_v$ denotes the check-in behavior of the user u_i at the location p_j, and each edge $\langle u_i, p_j\rangle \in E_v$ has a weight $(h_{\text{it}}, t_{\text{span}})$, where h_{it} denotes the number of visits paid by the user u_i to the location p_j and t_{span} denotes the time of the latest visit paid by the user u_i to the location p_j. $\langle p_k, p_j\rangle \in E_t$ indicates a trajectory that a user checked in at the location p_k and then checks in at the location p_j. E_t denotes the set of directed edges between two locations. An edge exists between two locations if and only if a user has once checked in at the two locations and the time interval between the two visits is less than a threshold, which is set to one week in this paper. Each edge $\langle p_k, p_j\rangle \in E_t$ has a weight $(h_{\text{it}E}, t_{\text{span}E})$, where $h_{\text{it}E}$ denotes the number of times that the locations p_k and p_j are visited sequentially and the conditions are satisfied; $t_{\text{span}E}$ denotes the latest time that the locations p_k and p_j are visited sequentially and the conditions are satisfied. The problem of recommending location for the user u_i can be described as the problem of estimating the correlation between the target user vertex and the location vertexes that have no link before in the user-location bipartite graph.

From the temporal aspect of view, the graph is varying with time. Figure 6 shows graph G in time t_1 and t_2 ($t_1 < t_2$). In time t_2, there happens a check-in between user u_n and location p_k. It produces an edge between them. Thus, there is a common trajectory between user u_i and u_n, that is, edge $\langle p_j, p_k\rangle$.

In order to obtain two users' common locations and repeated check-in times quickly, this paper presents a method named GraPA based on the message propagation and aggregation on the graph to determine the common trajectory and repeated times.

Input: Graph $G = \langle V, E\rangle$
Output: Message list for each vertex in Graph G: List $[(v\text{id}, L_i)]$
(1) **for** each $v_i \in V$ **do** /∗*Initialization*∗/
(2) $\quad L_i = v\text{id}_i$
(3) **for** each $\langle v_i, v_j\rangle \in E$ **do** /∗*Message Propagation*∗/
(4) $\quad L_j \leftarrow L_i$
(5) **for** each $v_i \in V$ **do** /∗*Message Aggregation*∗/
(6) $\quad$ **for** $k = 1 \cdots m$ **do**
(7) $\quad\quad$ **if** $k\langle\,\rangle i$ **then**
(8) $\quad\quad\quad L_i = L_i \cup L_k$
(9) **return** List $[(v\text{id}, L_i)]$

Algorithm 1: GraPA: message propagation and aggregation in graph.

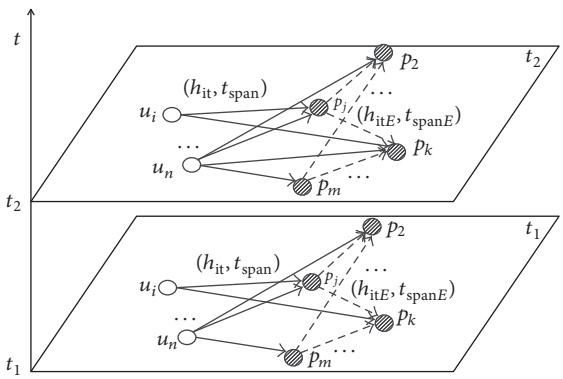

Figure 6: User-location bipartite graph with temporal information.

There are three states of vertices in a graph in GraPA method. The first state of the vertex is nonactivated, which is denoted as S_0. The second state of the vertex is activated, which is denoted as S_1, in which the vertex is propagating the message. The third state is denoted as S_2, in which the vertex is aggregating the received messages. Each vertex vi of the graph needs to maintain a message. The message of vertex v_i can be denoted as $vI = (v\text{id}_i, L_i)$, $v\text{id}_i$ is the ID of vertex v_i, and L_i is the set of vertex ID which would arrive at the vertex v_i during the message propagating. L_i is initialized as $v\text{id}_i$ itself. The states of all of the vertices alter among the three states according to time slot, so as to do message propagation and aggregation. The process of GraPA is depicted as Algorithm 1.

4. User Similarity Calculation

After the selected user check-in data subset is clustered into the regional data, the similar user set is extracted, and the user-location bipartite graph is constructed, we need to compute the similarity between target user and similar users. The point data and trajectory data in the user check-in data are jointly taken into account while computing the similarity.

4.1. Exploiting Spatio-Temporal Point Data. Each user has his/her preferences. The basic idea of the user-based collaborative filtering algorithm is the observation that the level of interuser similarity increases with the number of locations registered by the two users. The difference in the number of visits paid by the two users to the same location is taken into account in this paper, and the higher the difference, the lower the level of interuser similarity. Meanwhile, the check-in time is also considered. For two users who have once checked in at the same location, the longer the interval between their visits to the location, the lower the level of interuser similarity. Based on these observations, the interuser similarity can be computed as Formula (2):

$$\text{sim}_{\text{point}}\left(u_i, u_j\right) = \frac{-\sum_{i\in(P_{u_i}\cap P_{u_j})} th\left(\log\left(t_{\text{span}}(i) - \delta\right)\left(h_{\text{diff}}(i) - \beta\right)\right)}{\left|P_{u_i} \cap P_{u_j}\right|}, \quad (2)$$

where P_{u_i} and P_{u_j} denote the set of locations once checked in by the target user u_i and the similar user u_j in the subset D_1', respectively; $t_{\text{span}}(i)$ denotes the time interval between the latest visits of the target user u_i and the similar user u_j to the same location i, δ denotes the preset threshold of time interval between visits in millisecond, $h_{\text{diff}}(i)$ denotes the difference in the number of visits paid by the target user u_i and the similar user u_j to the same location i, and β is the preset largest threshold of the difference in the number of visits. In this paper, the interuser similarity $\text{sim}_{\text{point}}(u_i, u_j)$ ranges from −1 to 1. The lower the value of $t_{\text{span}}(i)$ and $h_{\text{diff}}(i)$, the more closer the value of $\text{sim}_{\text{point}}(u_i, u_j)$ to 1. This means the interuser similarity is higher. The higher the value of $t_{\text{span}}(i)$ and $h_{\text{diff}}(i)$, the more closer the value of $\text{sim}_{\text{point}}(u_i, u_j)$ to −1. This means the interuser similarity is smaller. If the value of $t_{\text{span}}(i)$ is larger than the constant δ or the value of $h_{\text{diff}}(i)$ is larger than the threshold β, the data of this location belongs to negative feedback and the interuser similarity $\text{sim}_{\text{point}}(u_i, u_j)$ is less than 0.

4.2. Exploiting User Check-In Trajectories. The trajectory in the user-location check-in data consists of a consecutive series of locations. It also includes the check-in time of different user locations, which is helpful in analyzing the mobility pattern of users at different time. The trajectory data is incorporated into the calculation of user similarity to

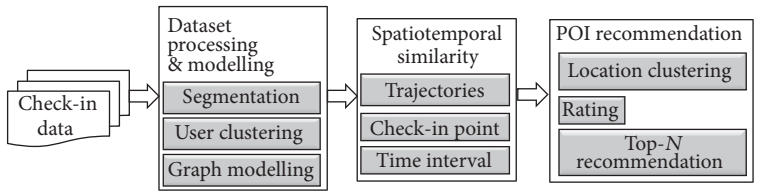

Figure 7: LGP-CF algorithm flowchart.

improve accuracy. If many of the trajectory data of one user is the same as the other user, it means that their mobility pattern is very similar and the interuser similarity is very high. Based on user trajectory data, the interuser similarity can be computed as Formula (3):

$$\mathrm{sim}_{\mathrm{traj}}\left(u_i, u_j\right) = th\left(\sum_{m,n\in(P_{u_i}\cap P_{u_j})} \mathrm{weight}\,(m,n)\right), \quad (3)$$

where weight(m,n) is equal to 0 or 1. When the following three conditions are satisfied,

(i) the locations $m,n \in (P_{u_i} \cap P_{u_j})$,

(ii) users u_i and u_j visit the locations m and n in the same sequence,

(iii) the time interval between the visits of the same user to the two places is shorter than the threshold β, and we have weight$(m,n) = 1$.

Given a target user and the similar user set Sim$U$, after the calculation of the spatio-temporal data-based user similarity $\mathrm{sim}_{\mathrm{point}}(u_i,u_j)$ and the trajectory data-based user similarity $\mathrm{sim}_{\mathrm{traj}}(u_i,u_j)$, the final user similarity can be computed as Formula (4):

$$\mathrm{sim}\left(u_i,u_j\right) = \mathrm{sim}_{\mathrm{point}}\left(u_i,u_j\right) + \mathrm{sim}_{\mathrm{traj}}\left(u_i,u_j\right). \quad (4)$$

4.3. Rating for Recommendable Locations. The traditional collaborative filtering algorithm is characterized by data sparsity and operational inefficiency and does not take the selection of recommendable locations into account. While choosing the recommendable locations, we consider the mobility pattern of users and cluster the longitude and latitude of the location set P into two classes. Afterwards, the class which has more locations in common with the set of target user-registered locations is identified. And the set of locations in this class which have not been checked in by the target user is regarded as the set of recommendable locations for the target user. The rating of recommendable location p_k for u_i is then calculated as in Formula (5). Finally, the top-N locations are recommended to the user.

$$r\left(u_i,p_k\right) = \frac{\sum_{u_j\in\mathrm{Sim}U}\mathrm{sim}\left(u_i,u_j\right)}{\left(1+\left(T-\mathrm{time}\left(u_j,p_k\right)\right)\right)}. \quad (5)$$

5. LGP-CF Algorithm and Its Parallel Design

5.1. LGP-CF Algorithm. Based on ideas above, we propose a new collaborative filtering-based spatio-temporal data-incorporated location recommendation algorithm LGP-CF for LBSN. Algorithm flowchart is shown in Figure 7. The location check-in data of all users is divided into three dataset segments according to the time period. Next, the dataset corresponding to the recommendation request time of the target user is selected. The subsequently constructed cluster of regional data is used to obtain the set of users similar to the target user. Then we model the filtered subset D_1' as a user-item (location) bipartite graph, over which we execute GraPA twice in order to find users' common locations and common trajectory between two common locations. These are the candidate similar users, locations, and trajectory of the target user. Then, the similarity between target user and each of the similar users is computed using the trajectory and point location. The locations are clustered using the longitude and latitude data. Finally, ratings are calculated and sorted using the location check-in time of similar users.

5.2. The Parallel Design of LGP-CF Algorithm. The pseudo code of LGP-CF is presented in Algorithm 2. The input of LGP-CF is the resilient distributed dataset (RDD) generated using the user-location check-in dataset and the parallel calculation framework Spark. RDD is not only an invariable partitioned set of records, but also a programming model of Spark. As in Hadoop, it submits the task at the two-stage of MapReduce and brings high delay between tasks. Different from Hadoop, Spark provides two RDD operations, that is, transformation and action. In Spark, a program actually constructs a directed acyclic graph (DAG) that consists of several interdependent RDDs. Various RDD operations are performed by submitting DAG as a task to Spark for execution. Hence, Spark tasks do not need to wait for each other, thereby improving the ability to process iterative data. Note that the data associated with each iteration of Spark is stored in the memory. This enables Spark to gain enormous performance improvement over Hadoop [24].

Each step of LGP-CF is parallelized and the data throughout the graph can thus be processed in a parallel manner. The first and second steps of Algorithm 2 are detailed here.

(1) In the first step, RDD needs to be converted into (user, location, latitude, longitude, hour, weekday) before the segmentation of the user-location check-in dataset according to time property, where hour denotes the hour of a day in the check-in time and weekday denotes the day of a week in the check-in time. Next, the converted RDD should be filtered based on the property (hour, weekday). RDD of the user check-in data subset that corresponds to the current time is obtained in this way.

(2) In the second step, RDD of the user check-in data subset needs to be converted to the user property RDD,

Input: user location check-in dataset: D; target user ID:TID.
Output: Recommended location list: List $[I_{\text{id}}]$
(1) Initialisation: $D' \leftarrow$ dataSetSplit (D); /$*D'$ *is the subset of check-in dataset D according to the target user's request time.*$*$/
(2) $U' = \{\overrightarrow{v_1}, \overrightarrow{v_2}, \ldots, \overrightarrow{v_n}\} \leftarrow$ userDataModel (D'); /$*$*Construct users property vector exploiting the regional data.*$*$/
(3) Sim$U \leftarrow k$MeansFilter (U'); /$*$*Clustering similarity users as* Sim$U$ *set.*$*$/
(4) $G = \langle V, E \rangle \leftarrow$ graphBuild (D_1'); /$*$*Model the filtered subset* D_1' *as a user-item (location) bipartite graph.*$*$/
(5) **for** $i = 0$ to 1 **do**
(6) GraPA (G); /$*$*traverse the graph to find candidate similar users, locations and trajectory*$*$/
(7) **for** each $u_i \in$ Sim$U$ **do**
(8) $\text{sim}(u_i, u_{\text{TID}}) = \text{sim}_{\text{point}} + \text{sim}_{\text{traj}}$
(9) $P' \leftarrow k$MeansLocation (P); /$*$*Clustering location to select candidate recommendable locations.*$*$/
(10) **for** each $p_k \in P'$ **do**
(11) **for** each $u_i \in$ Sim$U$ **do**
(12) $r(u_{\text{TID}}, p_k) = r(u_{\text{TID}}, p_k) + \text{sim}(u_i, u_{\text{TID}})/(1 + (T - \text{time}(u_{\text{TID}}, p_k)))$
(13) List$[I_{\text{id}}] \leftarrow$ sortByRating $(r_{pi}, r_{pj}, \ldots, r_{pk})$; /$*$*Top-N recommendation.*$*$/
(14) **return** List$[I_{\text{id}}]$

Algorithm 2: LGP-CF in LBSN.

that is, (u_i, lMaxLng, lMaxLat, lNearLng, lNearLat, lDistLng, lDistLat, lMaxWeek, lMaxDay), in order to construct the set of user properties. While constructing user property RDD, we need to first map the user check-in data subset RDD into ((user, location, longitude, latitude), 1) and name it user check-in RDD. Next, we compute the number of times that the user checked in at each of the registered locations through the key-based value processing operation reduceByKey. The combineByKey and mapping operations are performed to determine the location with the largest number of check-in times and the most registered location RDD (user, (lMaxLng, lMaxLat)). Afterwards, the mapping operation is done to convert the user check-in RDD into (user, longitude, latitude). Join and mapping operations are performed on it and the most registered location RDD, computing RDD of the distance between user check-in location and the most registered location. According to the distance property, we choose RDD of the location closest to the most registered location (user, lNearLng, lNearLat) and RDD of the location furthest from the most registered location (user, lDistLng, lDistLat). Similarly, the mapping operation, key-based value processing reduceByKey operation, and the clustering combineByKey operation are performed to determine RDD of the most frequent check-in hour in a day and RDD of the most frequent check-in day of a week. Finally, the join and mapping operations are performed to connect the most registered location RDD, RDD of the location closest to the most registered location, RDD of the location furthest from the most registered location, RDD of the most frequent check-in hour in a day, and RDD of the most frequent check-in day of a week. In this way, we finally obtain the user property RDD.

Each step of the Spark-based recommendation algorithm is parallelized and the calculation result of each step is stored in the buffer. After all tasks associated with the current step are completed, the buffered calculation result will be passed to the next step, resulting in fewer access to the disk, higher job execution efficiency, and improved algorithm performance.

6. Experiments and Evaluation

Experiment is conducted in this section to evaluate the recommendation performance of the proposed algorithm. Large-scale LBSN datasets from Gowalla and Foursquare are adopted in the experiment to evaluate algorithm performance. As in Section 3, the distribution of the number of user check-in times and the number of user check-in locations has been analyzed. And impact of different dataset segmentation on the recommendation results was discussed. In this section, LGP-CF is implemented and compared with other methods in the real-world physical cluster environment.

6.1. Experimental Environment. We use 6 servers in the experiment to build a cloud cluster. The Server OS is 64-bit Ubuntu14.04, cluster management platform is Spark1.1.0, and each server node includes a 4-core CPU and 8 GB memory. One server is configured as master and the other five as slave nodes. LGP-CF and other compared algorithms are implemented in a parallel manner in Spark.

6.2. Evaluation Results. Dataset segmentation of different time periods is used to evaluate the performance of the proposed algorithm in the experiment. Performance metrics include the predicted root-mean-square error (RMSE), precision, and recall.

Figures 8 and 9 compare precision and recall of LGP-CF on datasets of different time periods. From the two figures, it can be seen that the user check-in time concentrates in working days (after hours). Accuracy and recall of LGP-CF are very desirable. But algorithm performance is mediocre for time periods with a small number of user check-in times.

Based on this observation, we choose to compare LGP-CF with other algorithms on after-hour periods of working days. Because LGP-CF incorporates the spatio-temporal information, the traditional collaborative filtering algorithm L-CF is selected as a baseline algorithm for comparison. The aim is to

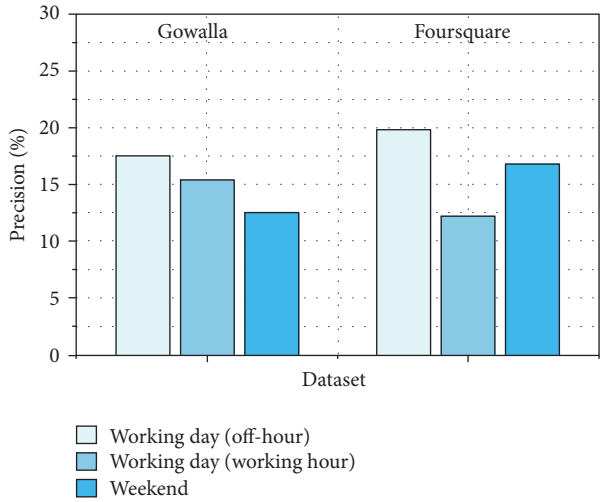

Figure 8: The precision in different periods.

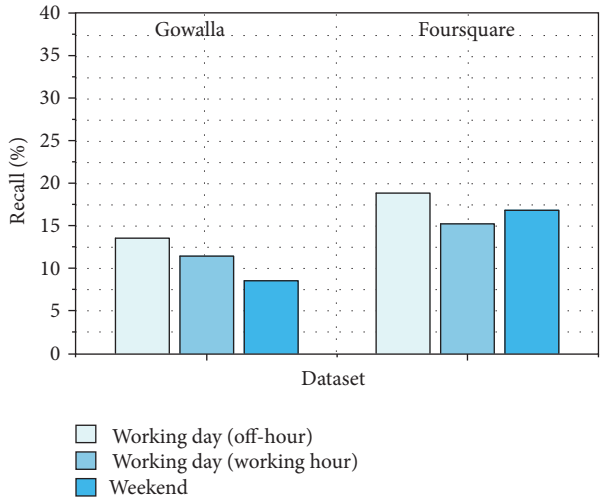

Figure 9: The recall in different periods.

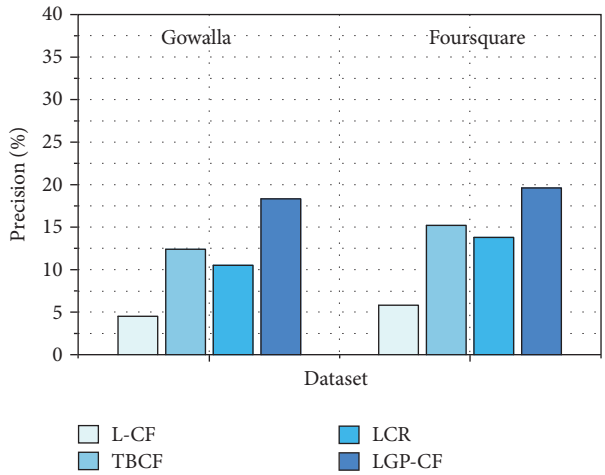

Figure 10: The precision comparison of different algorithms.

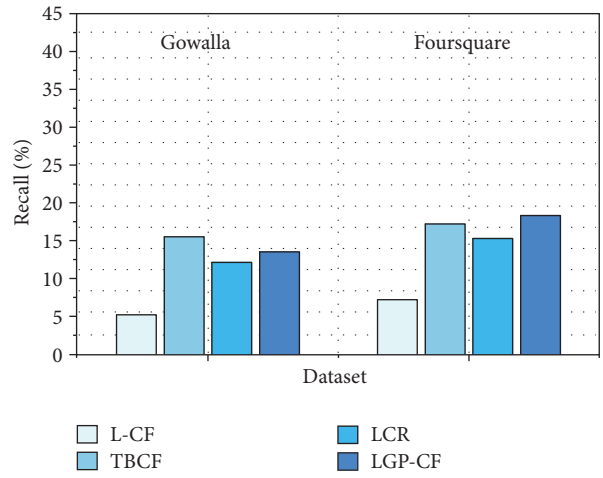

Figure 11: The recall comparison of different algorithms.

study the impact of incorporated spatio-temporal information on recommendation results. Afterwards, the time-based collaborative filtering algorithm TBCF is chosen to study the impact of combining temporal and spatial information on recommendation results. Finally, the LCR algorithm based on clustering of regional similarity is chosen to study the impact of combining spatio-temporal information with the clustering method on recommendation results.

Figures 10 and 11 show the comparison of all algorithms on the Gowalla and Foursquare datasets.

The comparison in Figures 10 and 11 indicates that LGP-CF is superior to L-CF in terms of precision and recall. This proves that LGP-CF achieves enormous performance gain in precision and recall, compared with the traditional collaborative filtering-based location recommendation algorithm for LBSN. While maintaining recall, LGP-CF produces higher accuracy than TBCF, because it incorporates both spatial and temporal information into recommendation, while TBCF only exploits the temporal information in its recommendation. Meanwhile, comparison between LGP-CF with LCR also indicates that combining spatio-temporal information with the clustering method enables LGP-CF to achieve improvements in precision and recall.

7. Conclusions

In addition to processing a large amount of data, existing LBSN location recommendation algorithms are used to meet various user needs. However, most of these methods are not accurate and efficient enough to make high-quality recommendation. In this paper, a new collaborative filtering-based spatio-temporal data-incorporated recommendation algorithm LGP-CF is proposed. User-location check-in data is divided according to time periods. The dataset that corresponds to user recommendation request time is then selected to reduce the amount of data that needs to be computed. Regional data associated with user mobility ranges is used to cluster users, obtain the set of similar users, and narrow the scope of choices of similar users. The selected user-location

check-in data subsets are modeled using the user-location bipartite graph. Therefore, the common visiting locations and the number of visits can be determined for two users by retrieving location edges in the directed graph. Afterwards, the trajectory data and point data corresponding to the set of similar users are used to compute the similarity between the target user and each of the similar users. The locations are then clustered using longitude and latitude data in order to obtain the set of recommendable locations accurately and reliably. Finally, the rating of a location is computed using the time of visits paid by similar users to it. Recommendation accuracy is improved in this way. Experiments on the real-world physical cluster are performed to compare with other LBSN recommendation algorithms. Results demonstrate superiority of LGP-CF in terms of precision and recall.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant no. 2016YFB1200100 and the Fundamental Research Funds for the Central Universities (2017JBM024).

References

[1] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in location-based social networks: a survey," *GeoInformatica*, vol. 19, no. 3, pp. 525–565, 2015.

[2] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "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, KDD 2014*, pp. 831–840, USA, August 2014.

[3] H. Gao, J. Tang, X. Hu, and H. Liu, "Modeling temporal effects of human mobile behavior on location-based social networks," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM 2013*, pp. 1673–1678, USA, November 2013.

[4] X. Liu, Y. Liu, and X. Li, "Exploring the context of locations for personalized location recommendations," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 1188–1194, New York, NY, USA, July 2016.

[5] H. Gao and H. Liu, "Data Analysis on Location-Based Social Networks," *Mobile Social Networking*, pp. 165–194, 2014.

[6] H. Gao, J. Tang, and H. Liu, "Personalized location recommendation on location-based social networks," in *Proceedings of the 8th ACM Conference on Recommender Systems, RecSys 2014*, pp. 399-400, USA, October 2014.

[7] O. Khalid, M. U. S. Khan, S. U. Khan, and A. Y. Zomaya, "OmniSuggest: A ubiquitous cloud-based context-aware recommendation system for mobile social networks," *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 401–414, 2014.

[8] H. Abdel-Fatao, J. Li, and J. Liu, "Unifying spatial, temporal and semantic features for an effective GPS trajectory-based location recommendation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9093, pp. 41–53, 2015.

[9] S. Zhao, I. King, and M. R. Lyu, "A Survey of Point-of-interest Recommendation in Location-based Social Networks," https://arxiv.org/abs/1607.00647, 2016.

[10] M. Huanyu, L. Zhen, W. Fang, and X. Jiadong, "Towards Efficient Collaborative Filtering Using Parallel Graph Model and Improved Similarity Measure," in *Proceedings of the 18th IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, pp. 182–189, Sydney, Australia, December 2016.

[11] R. Levin, H. Abassi, and H. Uzi Cohen, "Guided walk: A scalable recommendation algorithm for complex heterogeneous social networks," in *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016*, pp. 293–300, Boston, Mass, USA, September 2016.

[12] L. Yao, Q. Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, and Q. He, "Context-aware point-of-interest recommendation using Tensor Factorization with social regularization," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2015*, pp. 1007–1010, Chile, August 2015.

[13] R. Baral and T. Li, "MAPS: A multi aspect personalized POI recommender system," in *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016*, pp. 281–284, USA, September 2016.

[14] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging Collaborative Filtering and Semi-Supervised Learning," in *Proceedings of the the 23rd ACM SIGKDD International Conference*, pp. 1245–1254, Halifax, NS, Canada, August 2017.

[15] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks," in *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing, SocialCom 2012 and the 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2012*, pp. 144–153, Netherlands, September 2012.

[16] Y. Zheng, "Trajectory data mining: an overview," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, article 29, 2015.

[17] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD International Conference*, pp. 25–34, ACM, August 2014.

[18] C. Zhang and K. Wang, "POI recommendation through cross-region collaborative filtering," *Knowledge and Information Systems*, vol. 46, no. 2, pp. 369–387, 2016.

[19] M. Aamir, "Dynamicity in Social Trends towards Trajectory Based Location Recommendation," in *Proceedings of the International Conference on Smart Homes and Health Telematics*, pp. 86–93, Singapore, Singapore, 2013.

[20] S. Shang, K. Zheng, C. S. Jensen et al., "Discovery of path nearby clusters in spatial networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 6, pp. 1505–1518, 2015.

[21] Z. Jin, D. Shi, Q. Wu, H. Yan, and H. Fan, "LBSN Rank: personalized pagerank on location-based social networks," in *Proceedings of the the 2012 ACM Conference*, pp. 980–987, Pittsburgh, Penn, USA, September 2012.

[22] C. Cui, J. Shen, L. Nie, R. Hong, and J. Ma, "Augmented Collaborative Filtering for Sparseness Reduction in Personalized POI

Recommendation," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 5, pp. 71–93, 2017.

[23] C. Cheng, H. Yang, I. King, and M. R. Lyu, "A unified point-of-interest recommendation framework in Location-based social networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 1, article 10, 2016.

[24] R. S. Xin, D. Crankshaw, A. Dave et al., "GraphX unifying data-parallel and graph-parallel analytics," *Computer Science Databases*, 2014, https://arxiv.org/abs/1402.2394.