Chapter 8
Location-Based Social Networks: Users

Yu Zheng

Abstract In this chapter, we introduce and define the meaning of location-based social network (LBSN) and discuss the research philosophy behind LBSNs from the perspective of users and locations. Under the circumstances of trajectory-centric LBSN, we then explore two fundamental research points concerned with understanding users in terms of their locations. One is modeling the location history of an individual using the individual's trajectory data. The other is estimating the similarity between two different people according to their location histories. The inferred similarity represents the strength of connection between two users in a location-based social network, and can enable friend recommendations and community discovery. The general approaches for evaluating these applications are also presented.

8.1 Introduction

8.1.1 Concepts and Definitions of LBSNs

A social network is a social structure made up of individuals connected by one or more specific types of interdependency, such as friendship, common interests, and shared knowledge. Generally, a social networking service builds on and reflects the real-life social networks among people through online platforms such as a website, providing ways for users to share ideas, activities, events, and interests over the Internet.

The increasing availability of location-acquisition technology (for example GPS and Wi-Fi) empowers people to add a location dimension to existing online social networks in a variety of ways. For example, users can upload location-tagged photos

Yu Zheng
Microsoft Research Asia, China
e-mail: yuzheng@microsoft.com

to a social networking service such as Flickr [2], comment on an event at the exact place where the event is happening (for instance, in Twitter [6]), share their present location on a website (such as Foursquare [3]) for organizing a group activity in the real world, record travel routes with GPS trajectories to share travel experiences in an online community (for example GeoLife [60, 57, 53, 61]), or log jogging and bicycle trails for sports analysis and experience sharing (as in Bikely [1] and [15]).

Here, a location can be represented in absolute (latitude-longitude coordinates), relative (100 meters north of the Space Needle), and symbolic (home, office, or shopping mall) form. Also, the location embedded into a social network can be a stand-alone instant location of an individual, like in a bar at 9pm, or a location history accumulated over a certain period, such as a GPS trajectory: "a cinema→a restaurant→a park→a bar."

The dimension of location brings social networks back to reality, bridging the gap between the physical world and online social networking services. For example, a user with a mobile phone can leave her comments with respect to a restaurant in an online social site (after finishing dinner) so that the people from her social structure can reference her comments when they later visit the restaurant. In this example, users create their own location-related stories in the physical world and browse other people's information as well. An online social site becomes a platform for facilitating the sharing of people's experiences.

Furthermore, people in an existing social network can expand their social structure with the new interdependency derived from their locations. As location is one of the most important components of user context, extensive knowledge about an individual's interests and behavior can be learned from her locations. For instance, people who enjoy the same restaurant can connect with each other. Individuals constantly hiking the same mountain can be put in contact with each other to share their travel experiences. Sometimes, two individuals who do not share the same absolute location can still be linked as long as their locations are indicative of a similar interest, such as beaches or lakes.

These kinds of location-embedded and location-driven social structures are known as location-based social networks, formally defined as follows:

A location-based social network (LBSN) does not only mean adding a location to an existing social network so that people in the social structure can share location-embedded information, but also consists of the new social structure made up of individuals connected by the interdependency derived from their locations in the physical world as well as their location-tagged media content, such as photos, video, and texts. Here, the physical location consists of the instant location of an individual at a given timestamp and the location history that an individual has accumulated in a certain period. Further, the interdependency includes not only that two persons co-occur in the same physical location or share similar location histories but also the knowledge, e.g., common interests, behavior, and activities, inferred from an individual's location (history) and location-tagged data.

In a location-based social network, people can not only track and share the location-related information of an individual via either mobile devices or desktop computers, but also leverage collaborative social knowledge learned from user-generated and location-related content, such as GPS trajectories and geo-tagged photos. One example is determining this summer's most popular restaurant by mining people's geo-tagged comments. Another example could be identifying the most popular travel routes in a city based on a large number of users' geo-tagged photos. Consequently, LBSNs enable many novel applications that change the way we live, such as physical location (or activity) recommendation systems [65, 63, 59, 50, 51, 58, 10] and travel planning [45, 46], while offering many new research opportunities for social network analysis (like user modeling in the physical world and connection strength analysis) [28, 39, 16, 20, 21, 19, 25, 44], spatio-temporal data mining [29, 47, 49, 42, 64], ubiquitous computing [55, 54, 52, 56, 48, 62], and spatio-temporal databases [35, 13, 12, 37, 14, 18].

8.1.2 Location-Based Social Networking Services

Existing applications providing location-based social networking services can be broadly categorized into three folds: geo-tagged-media-based, point-location-driven and trajectory-centric.

- *Geo-tagged-media-based*. Quite a few geo-tagging services enable users to add a location label to media content such as text, photos, and videos generated in the physical world. The tagging can occur instantly when the medium is generated, or after a user has returned home. In this way, people can browse their content at the exact location where it was created (on a digital map or in the physical world using a mobile phone). Users can also comment on the media and expand their social structures using the interdependency derived from the geo-tagged content (for example, in favor of the same photo taken at a location). Representative websites of such location-based social networking services include Flickr, Panoramio, and Geo-twitter. Though a location dimension has been added to these social networks, the focus of such services is still on the media content. That is, location is used only as a feature to organize and enrich media content while the major interdependency between users is based on the media itself.
- *Point-location-driven*. Applications like Foursquare and Google Latitude encourage people to share their current locations, such as a restaurant or a museum. In Foursquare, points and badges are awarded for "checking in" at venues. The individual with the most number of "check-ins" at a venue is crowned "Mayor." With the real-time location of users, an individual can discover friends (from her social network) around her physical location so as to enable certain social activities in the physical world, e.g., inviting people to have dinner or go shopping. Meanwhile, users can add "tips" to venues that other users can read, which serve as suggestions for things to do, see, or eat at the location. With this kind of service, a venue (point location) is the main element determining the in-

terdependency connecting users, while user-generated content such as tips and badges feature a point location.

- *Trajectory-centric.* In a trajectory-centric social networking service, such as Bikely, SportsDo, and Microsoft GeoLife, users pay attention to both point locations (passed by a trajectory) and the detailed route connecting these point locations. These services do not only tell users basic information, such as distance, duration, and velocity, about a particular trajectory, but also show a user's experiences represented by tags, tips, and photos for the trajectory. In short, these services provide "how and what" information in addition to "where and when." In this way, other people can reference a user's travel/sports experience by browsing or replaying the trajectory on a digital map, and follow the trajectory in the real world with a GPS-phone.

Table 8.1 provides a brief comparison among these three services. The major differences between the point-location-driven and the trajectory-centric LBSN lie in two aspects. One is that a trajectory offers richer information than a point location, such as how to reach a location, the temporal duration that a user stayed in a location, the time length for travelling between two locations, and the physical/traffic conditions of a route. As a result, we are more likely to accurately understand an individuals behavior and interests in a trajectory-centric LBSN. The other is that in a point-location-driven LBSN users usually share their real-time location while the trajectory-centric more likely delivers historical locations as users typically prefer to upload a trajectory after a trip has finished (though it can be operated in a continuously uploading manner). This property could compromise some scenarios based on the real-time location of a user, however, it reduces to some extent the privacy issues in a location-based social network. In other words, when people see a users trajectory the user is no longer there.

Table 8.1 Comparison of different location-based social networking services

LBSN Services	Focus	Real-time	Information
Geo-tagged-media-based	Media	Normal	Poor
Point-location-driven	Point location	Instant	Normal
Trajectory-centric	Trajectory	Relatively Slow	Rich

Actually, the location data generated in the first two LBSN services can be converted into the form of a trajectory which might be used by the third category of LBSN service. For example, if we sequentially connect the point locations of the geo-tagged photos taken by a user over several days, a sparse trajectory can be formulated. Likewise, the check-in records of an individual ordered by time can be regarded as a low-sampling-rate trajectory. However, due to the sparseness, i.e., the distance and time interval between two consecutive points in a trajectory could be very big, the uncertainty existing in a single trajectory from the first two services is increased. Aiming to put these trajectories into trajectory-centric LBSN services, we need to use them in a collective and collaborative way.

The following sections will pay closer attention to trajectory data, which is the most complex data structure to be found in the three LBSN services, and provides the richest information. If it is handled well, other data sources become easier to deal with. Moreover, as mentioned above, location data can be converted into a trajectory on many occasions. Consequently, some methodologies designed for trajectory data can be employed by the first two LBSN services.

8.1.3 Research Philosophy of LBSN

User and location are two major subjects closely associated with each other in a location-based social network. As illustrated in Fig. 8.1, users visit some locations in the physical world, leaving their location histories and generating location-tagged media content. If we sequentially connect these locations in terms of time, a trajectory will be formulated for each user. Based on these trajectories, we can build three graphs: a location-location graph, a user-location graph, and a user-user graph.

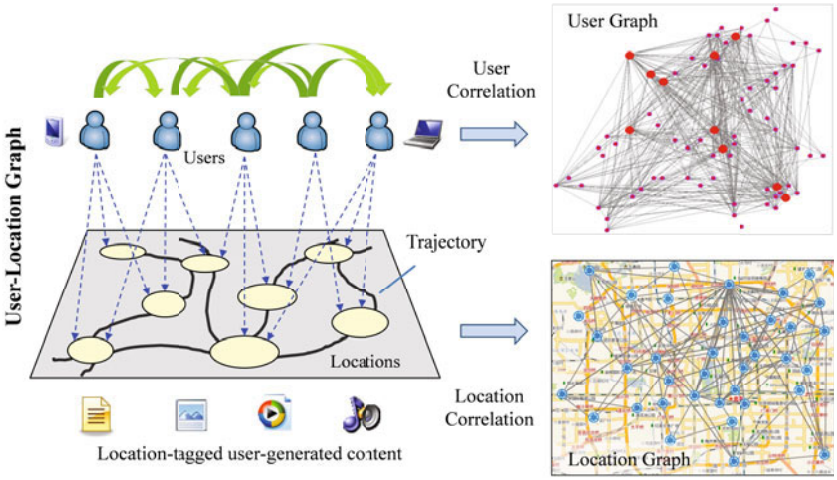

Fig. 8.1 Research philosophy of a location-based social network

In the location-location graph (demonstrated in the bottom-right of Fig. 8.1), a node (a point on the graph) is a location and a directed edge (a line on the graph) between two locations indicates that some users have consecutively traversed these two locations during a trip. The weight associated with an edge represents the correlation between the two locations connected by the edge.

In the user-location graph (depicted in the left part of Fig. 8.1), there are two types of nodes: users and locations. An edge starting from a user and ending at a location indicates that the user has visited this location, and the weight of the edge can indicate the number of visits.

In the user-user graph (shown in the top-right of Fig. 8.1), a node is a user and an edge between two nodes consists of two folds. One is the original connection between two users in an existing social network like Twitter. The other is the new interdependency derived from their locations, e.g., two users have visited the same location, or similar types of places, in the real world over a certain number of visits. The latter information, initially inferred from a user's locations, can be transferred to the former through a recommendation mechanism. In other words, we can recommend users to an individual based on the inferred interdependency. Once the individual accepts the recommendation, the relationship switches from the second category to the first.

Using these graphs, we can understand users and locations respectively, and explore the relationship between them. Though the research topics are listed individually from the perspective of users and locations as follows, these two subjects have a mutually reinforcing relationship that cannot be studied alone:

1) Understanding users: Here, we aim to understand users based upon their locations.
   - *Estimate user similarity* [28, 16]: An individual's location history in the real world implies, to some extent, her interests and behaviors. Accordingly, people who share similar location histories are likely to have common interests and behavior. The similarity between users inferred from their location histories can enable friend recommendations, which connect users with similar interests even when they may not have known each other previously [63], and community discovery that identifies a group of people sharing common interests.
   - *Finding local experts in a region* [65]: With users' locations, we are able to identify the local experts who have richer knowledge about a region than others. Their travel experiences, e.g., the locations where they have been, are more accountable and valuable for travel recommendation. For instance, local experts are more likely to know about high-quality restaurants than some tourists.
   - *Community discovery* [39, 25]: Using the similarity inferred from users' locations, we can cluster these users into groups in which users share common interests like visiting museums. Consequently, an individual can easily initiate a group activity, such as hiking or purchasing tickets at a group price, by sending an invitation to the appropriate users in a social site.
2) Understanding locations: Here, we focus on understanding locations based upon user information.
   - *Generic travel recommendations*:
     - Mining the most interesting locations [65]: Finding the most interesting locations in a city as well as the travel sequences among these locations is a general task that a tourist wants to fulfill when traveling to an unfamiliar city. Location-based social networks provide us with the opportunity to identify such information by mining a large number of users' location histories (represented by trajectories). Refer to Section 9.2.1.

  - Itinerary planning [45, 46]: Sometimes, a user needs a sophisticated itinerary conditioned by the user's travel duration and departure place. The itinerary could include not only stand-alone locations but also detailed routes connecting these locations and a proper schedule, e.g., the typical time of day that most people reach the location and the appropriate time length that a tourist should stay there. Planning a trip in terms of the collective knowledge learned from many people's trajectories is an interesting research topic. Refer to Section 9.2.2.
  - Location-activity recommender [51]: This recommender provides a user with two types of recommendations: 1) The most popular activities that can be performed in a given location and 2) the most popular locations for conducting a given activity, such as shopping. These two categories of recommendations can be mined from a large number of users' trajectories and location-tagged comments. Refer to Section 9.2.3.

- *Personalized travel recommendations*
  - User-based collaborative filtering [63]: In this scenario, the similarity between each pair of users (introduced above) is incorporated into a collaborative filtering model to conduct a personalized location recommendation system, which offers locations matching an individual's preferences. The general idea behind collaborative filtering [23, 30] is that similar users vote in a similar manner on similar items. Thus, if similarity is determined between users and items, predictions can be made about a user's potential ratings of those items. For instance, if we know user A and B are very similar (in terms of their location histories), we can recommend the locations where user A has already been to user B and vice versa. Refer to Section 9.3.2 for details.
  - Location-based collaborative filtering [59, 58]: User-based collaborative filtering is able to accurately model an individual's behavior. However, it suffers from the increasing scale of users (in a real system) since the model needs to calculate the similarity between each pair of users. To address this issue, location-based collaborative filtering is proposed. This model regards a physical location as an item and computes the correlation between locations based on the location histories of the users visiting these locations. Given the limited geographical space (i.e., the number of locations is limited), this location-based model is more practical for a real system. The main challenge to the location-based model is how to embody an individual's behavior which is the advantage of a user-based model. Refer to Section 9.3.3 for details.

- *Events discovery from social media* [29, 27]
  Quite a few projects aim to detect anomalous events, such as concerts, traffic accidents, sales promotions, and festivals, using media (such as geo-tagged photos and tweets) posted by a large number of users in a location-based social network. Intuitively, people witnessing such an event would post a

considerable amount of media (e.g., tweets) in the location where the event occurs. By grouping and mining media that co-occurs in particular locations, we can get a sense of geo-social events automatically.

Actually, the problems that traditional social networks have exist in location-based social networks, and become more challenging due to the following reasons:

- The graph representing a location-based social network is heterogeneous, consisting of at least two types of nodes (user and location) and three kinds of links (user-user, location-location, and user-location). Or, we can say there are at least three tightly associated graphs modeling a LBSN (as mentioned previously). If it is a trajectory-centric LBSN, trajectories can be regarded as another kind of node in the social network; so do geotagged videos and photos. Location is not only an additional dimension of the user, but also an important object in a LBSN. Under the circumstances, determining the connecting strength between two users in a LBSN needs to involve the information from the other graphs, such as the linking structure of user-location and location-location, besides that of users (refer to Fig. 8.1).
- Location-based social networks are constantly evolving at a faster pace than traditional social networks, in both social structure and properties of nodes and links. Though academic social networks are also heterogeneous with authors, conferences, and papers, its evolves at a much slower speed than LBSNs do. For example, it is much easier to add a new location to a LBSN (by check-in) than launching a new conference or publishing a paper. That is, the number of nodes in a LBSN increases faster than an academic social network. Also, it is common for users to visit locations (e.g., restaurants and shopping malls) they have never been before. However, researchers will not constantly attend new conferences. Thus, the linking structure of a LBSN evolves much faster than an academic social network. Furthermore, the properties of nodes and links in a LBSN evolve more quickly than in an academic social network. A user can become a travel expert in a city after visiting many interesting locations over several months, while a researcher needs years before becoming an expert in a research area.
- A location has unique features beyond that of other objects in a social network. Besides general linking relationship between locations, the hierarchical and sequential properties of locations are unique. A location can be as small as a restaurant or as big as a city. Locations with different granularities formulate hierarchies between them. For example, a restaurant belongs to a neighborhood, and the neighborhood pertains to a city. Further, the city will belong to a county and a country, and so on. Using different granularities, we will obtain different location graphs even given the same trajectory data. This hierarchical property does not hold in an academic social network as a conference never belongs to others. Regarding the sequential property, each link between two locations is associated with temporal and directional information. Moreover, these links can construct a sequence carrying a particular semantic meaning, e.g., a popular travel route.

There are other important research points in location-based social networks. For example, from the perspective of data management, streaming databases and indexing user-generated location data are vital. Also, user privacy in location-based social networks deserves to be further studied. As these topics have been discussed extensively in other chapters, they will not be covered here.

So far, there is no dedicated conference for researchers and professionals to share the research into LBSNs. While people submit LBSN-related papers to a number of conferences such as WWW, Ubicomp, and ACM GIS, ACM SIGSPATIAL Workshop on Location-Based Social Networks provides a dedicated international forum for LBSN researchers and practitioners from academia and industry to share their ideas, research results, and experiences. This workshop was launched in 2009 and has been in conjunction with ACM SIGSPATIAL GIS conference from 2009 to 2011.

8.2 Modeling Human Location History

8.2.1 Overview

To carry out the above-mentioned research, it is first necessary to model the location history of an individual from raw sensor data, such as GPS readings. The methods presented in most literature [8, 24] solely pay attention to detecting significant places from the sensor data, without considering the social computing among different users. That is, they do not study how to compare different users' location histories when modeling the location data of multiple persons. Since 2008, a series of publications [65, 63, 59, 10, 28, 39] proposed a systematical solution for this problem, following the paradigm of "sensor data→ geospatial locations (significant places)→semantic meanings (e.g., restaurants)." Beyond the related methods, this solution has the following two advantages: 1) Modeling the location history of an individual and that of many users simultaneously, thereby making different users' location histories comparable and computable; 2) Modeling an individual's travel in geospatial and semantic spaces respectively, allowing deeper understanding of the individual's behavior and interests.

Given these advantages, this solution will be further introduced in later sections. This paradigm is further illustrated using Fig. 8.2 as an example, in which two users visited some locations and created two trajectories, Tr_1 and Tr_2, respectively.

Directly measuring these two users' location histories based on the GPS readings (denoted by points) is difficult for two reasons. First, the raw sensor readings of these two users are different even if they were visiting the same location, such as A and C. This is caused by the intrinsic positioning error of a location-acquisition technology and the randomness of people's movement (e.g., people exit a building from different gates). Second, defining a proper distance threshold (e.g. 100 meters) is often arbitrary in determining whether two readings belong to the same location. If

we regard two points with a Euclidian distance smaller than 100 meters as readings from the same location, why do those points having a distance of 101 meters not also pertain to the same location? To address this issue, we need to convert a user's location history from sensor readings into a sequence of comparable locations in the geographic spaces, for instance, $Tr_1 : A \rightarrow C$, and $Tr_2 : A \rightarrow B \rightarrow C \rightarrow D$. Note that the focus is on the significant places like A and B where an individual carried out some meaningful behavior (reflecting her interests), such as shopping and watching a movie, instead of some points generated when an individual passes by a location like a crossroad without taking any essential action. This process will be discussed in more detail in Section 8.2.2.

However, knowing an individual's movement in the geographic spaces is not enough to understand the individual's interests. The semantic meaning of a physical location, e.g., a shopping center, will bring richer knowledge and context to explore a user's behavior. Given this reason, a user's trajectory is further converted from "$Tr_1 : A \rightarrow C$" to "a lake$\rightarrow$ a shopping center," thereby modeling the user's location history in terms of semantic spaces. Refer to Section 8.2.3 for details.

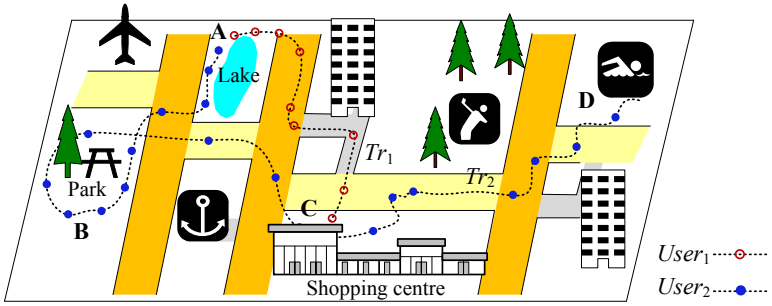

Fig. 8.2 Modeling the location history of a user from sensor data

8.2.2 Geospatial Model Representing User Location History

In this section, a framework is proposed, called a hierarchical graph, to uniformly model each individual's location history in the geospatial spaces [63, 28]. The framework consists of the following three steps, which are further illustrated in Fig. 8.3 and respectively detailed in later sections.

1) Detect significant places: The stay points are determined, each of which denotes a geographic region where an individual stayed for a certain duration, from the trajectory data. As compared to a raw sensor reading, each stay point carries a particular semantic meaning, such as the shopping malls and restaurants visited by an individual. Refer to Section 8.2.2.1 for details.

2) Formulate a shared framework: All users' stay points are placed together into a dataset. Using a density-based clustering algorithm, this dataset is recursively clustered into several clusters in a divisive manner. Thus, similar stay points from various users are assigned to the same cluster, and the clusters on different layers represent locations (geographical regions) of different granularities. This structure of clusters, referred to as a hierarchical framework, provides various users with a uniform framework to formulate their own graphs. Refer to the middle box shown at the bottom of Fig. 8.3.

3) Construct a personal location history: By projecting the individual location history onto the shared hierarchical framework, each user can build a personal directed-graph, in which a graph node is the cluster containing the user's stay points and a graph edge stands for the user's traveling sequence between these clusters (geographic regions). To simplify the problem, we do not differentiate between the diverse paths that a user created between two places (clusters).

In later sections, GPS logs are used as an exemplary trajectory to illustrate the methodology. Of course, this solution can be applied to other trajectory data sources, such as geo-tagged photos.

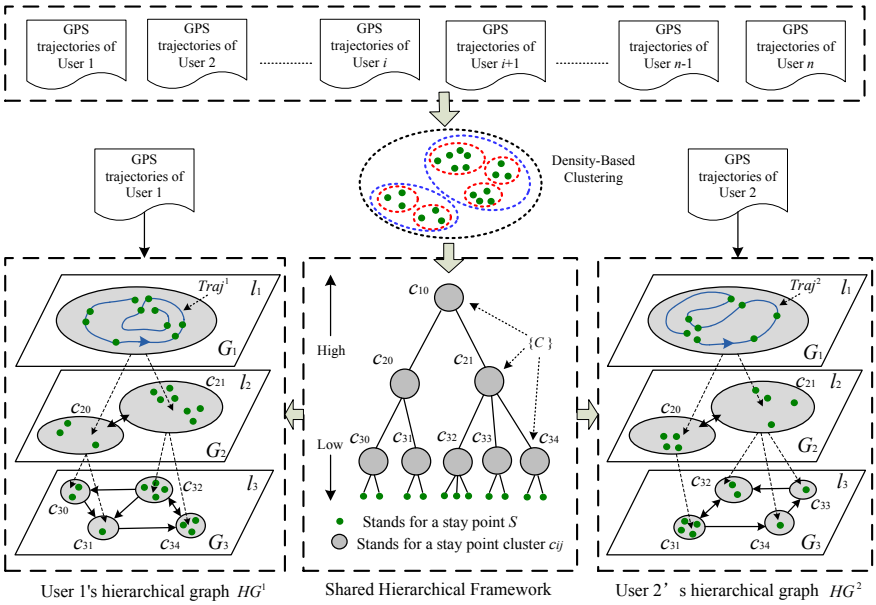

Fig. 8.3 Framework for modeling users location history in geographical spaces

8.2.2.1 Detecting Significant Places

A significant place denotes the location where an individual carried out some meaningful behavior (reflecting her interests), such as shopping, watching a movie, or visiting a museum. These significant places allow a better understanding of an individual's interests, thereby accurately computing the interdependency between different users. At the same time, other sensor readings outside of these significant places can be skipped, saving the computational load in a real system. Literature that introduces a method for detecting significant places includes [10, 28, 8, 24]. Basically, they share the idea of using a spatial and temporal constraint to delineate a location from a sequence of GPS coordinates. One representative method [28, 48] is selected and introduced below. Before going into detail, it is necessary to first define some terms that will be used in Chapters 8 and 9.

Definition 8.1 (GPS Trajectory). A GPS trajectory $Traj$ is a sequence of time-stamped points, $Tra = p_0 \rightarrow p_1 \rightarrow \cdots \rightarrow p_k$, where $p_i = (x, y, t), (i = 0, 1, \ldots, k)$; (x, y) are latitude and longitude respectively, and t is a timestamp. $\forall 0 \leq i \leq k, p_(i+1).t > p_i.t$.

Definition 8.2. $Dist(p_i, p_j)$ denotes the geospatial distance between two points p_i and p_j, and $Int(p_i, p_j) = |p_i.t - p_j.t|$ is the time interval between two points.

Definition 8.3 (Stay Point). A stay point s stands for a geographic region where a user stayed over a certain time interval. The extraction of a stay point depends on two scale parameters, a time threshold (τ) and a distance threshold (δ). Formally, given a trajectory, $Traj : p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, a single stay point s can be regarded as a virtual location characterized by a sub-trajectory $p_i \rightarrow \cdots \rightarrow p_j$, which satisfies the conditions that $\forall k \in [i, j)$, $Dist(p_k, p_(k+1)) < \delta$, $Int(p_i, p_j) > \tau$. Therefore, $s = (x, y, t_a, t_l)$, where

$$s.x = \sum_{k=i}^{j} p_k.x/|s|, \tag{8.1}$$

$$s.y = \sum_{k=i}^{j} p_k.y/|s|, \tag{8.2}$$

respectively stands for the average x and y coordinates of the stay point s; $s.t_a = p_i.t$ is the user's arriving time on s and $s.t_l = p_j.t$ represents the user's departure time.

Note that a stay point does not necessarily mean a user remains stationary in a location. Also, we do not expect to include the circumstance when an individual is stuck in a traffic jam or waiting for a traffic signal. Instead, we aim to detect the significant stays reflecting the semantic meanings of an individual's behavior and interests, which usually occur in the following two situations. One is when people enter a building and lose satellite signal over a time interval before coming back outdoors. Figure 8.4 A) shows an example in which an individual visited a shopping mall and stayed inside for a period of time. The other situation is when a

user exceeds a time limit at a certain geospatial area (outdoors). For instance, people strolling along a nice beach (refer to Fig. 8.4 B)), or being attracted by a landmark (See Fig. 8.4 C)) could generate a stay point.

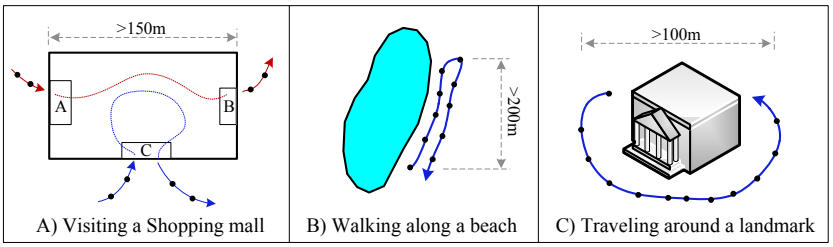

Fig. 8.4 Some examples of stay points

Figure 8.5 demonstrates the algorithm for stay point detection, using a trajectory $(p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_7)$. Overall, the stay point detection algorithm includes two operations: checking spatio-temporal constraint and expanding. As depicted in Fig. 8.5 B), p_1 and p_2 cannot formulate a stay point as $Dist(p_1, p_2)$ exceeds the corresponding threshold δ. Then, we move to p_2 and find that $Dist(p_2, p_3) < \delta$ and $Dist(p_2, p_4) < \delta$ while $Dist(p_2, p_5) > \delta$ (see Fig. 8.5 C)). If the time interval between p_2 and p_4 is larger than time threshold τ, the three points form a small cluster representing a stay point. However, they might not be the entire set of the points in this stay. Accordingly, we try to expand the stay point by continuously checking the distance between p_4 and the remaining points (p_5, p_6, p_7) in the trajectory. As depicted in Fig. 8.5 D), p_5 and p_6 are added into this stay point since they also meet the spatio-temporal constraints. Finally, we detect $(p_2 \rightarrow p_3 \rightarrow p_4 \rightarrow p_5 \rightarrow p_6)$ as a stay point because we cannot expand the cluster any further. That is, all the points in the cluster have a distance farther than δ to p_7.

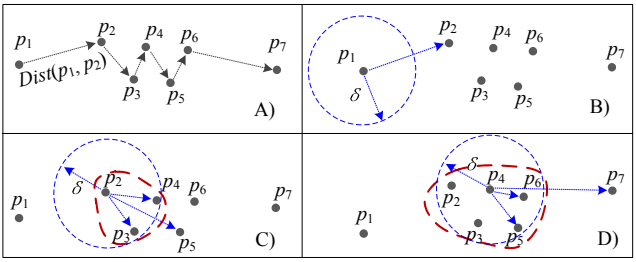

Fig. 8.5 An example of stay pints detection

At this point, some people might ask why not use an already existing clustering algorithm like DBSCAN to determine the stay points. There are two reasons for not doing so. On the one hand, as depicted in Fig. 8.4 A), if stay points are detected

by directly clustering raw GPS points, most significant places like shopping malls and restaurants will remain undetected. This is caused by the fact that GPS devices lose satellite signal indoors, i.e., few GPS points will be generated at those places. However, some places like intersections passed by many people will be identified as stay points. On the other hand, if we use an interpolation operation (to fill the lost GPS points), the computational load for clustering such a big dataset will be extremely heavy. For instance, a 2-hour stay will generate 720 points if we use 5-seconds as the interpolating frequency. The workload of clustering is impractical for a real system with an increasing number of users.

However, the selection of thresholds δ and τ for the algorithm is still not easy and depends on people's commonsense knowledge. For example, in the experiment of [28, 63], if an individual spent more than 15 minutes within a distance of 200 meters, the region is detected as a stay point. Although the aim is to represent each stay of a user as precisely as possible, we have to use a proper geo-region to specify an individual's stay for a number of reasons.

First, a strict region size, such as 20×20 meters, might be more capable of accurately identifying a business like a Starbucks visited by a user; however, it would cause many stays to remain undetected. As demonstrated in Fig. 8.4 A), a user could enter a shopping mall from Gate A while leaving the mall from Gate B (see the blue line). Given that a shopping mall could cover a 150×150 meter geo-region, the distance between the last GPS point before entering the mall and the first point after coming out from the mall could be larger than 150 meters; i.e., the user's stay at this shopping mall cannot be detected using a very small region constraint like 20 meters. Moreover, even if a user leaves the shopping mall from the same gate they entered, in most cases, the distance between the last GPS point before entering and the first point after coming out could be larger than 100 meters. Typically, GPS devices need some time to re-locate themselves after returning outdoors.

Second, a very small region constraint could cause the stays of people to be over-detected. As shown in Fig. 8.4 B) and C), multiple trivial stay points could be detected in one location when people stroll along a beach or wander around a landmark. The data does not align with a person's perceptions that she has only accessed one location (the beach or the landmark).

Third, these two parameters (200 meters, 15 minutes) are likely to exclude a situation where people wait for a signal at a traffic light, and can reduce to some extent the stay points caused by traffic jams, e.g., a traffic light does not normally last for 10 minutes.

8.2.2.2 Formulating a Shared Framework

After detecting the stay points from an individual's GPS trajectories, we can model the individual's location history with a sequence of stay points, which is defined as follows:

Definition 8.4 (Location histroy). Generally, location history is a record of locations that an entity visited in geographical spaces over an interval of time. In this book,

an individual's location history (LocH) is represented by a sequence of stay points (s) they visited with the corresponding arrival and departure times.

$$LocH = (s_1 \xrightarrow{\triangle t_1} s_2 \xrightarrow{\triangle t_2} \cdots \xrightarrow{\triangle t_{n-1}} s_n) \tag{8.3}$$

where $s_i \in S$ and $\triangle t_i = s_{i+1}.t_a - s_i.t_l$.

However, different people's location histories represented by a sequence of stay points are still inconsistent and incomparable even if they visited the same place. Besides the intrinsic positioning error of GPS sensors (as explained in Section 8.3.1), people accessing a location in a variety of ways, such as different directions, entrances, and exits, could also generate a variety of stay points in the location. Figure 8.4 A) gives an example where two users visit one building from two different gates and generate two different stay points.

To uniformly model each individual's location history, a shared framework that is formulated by hierarchically clustering all users' stay points is proposed, as formally defined in Definition 8.5.

Definition 8.5 (Shared Hierarchical Framework F). F is a collection of stay point-based clusters C with a hierarchy structure L. $F = (C, L)$, where $L = l_1, l_2, \ldots, l_n$ denotes the collection of layers of the hierarchy. $C = \{c_{ij} | 1 \leq i \leq |L|, 0 \leq j \leq |C_i|\}$, where c_{ij} denotes the jth cluster of stay points on layer $l_i (l_i \in L)$, and C_i is the collection of clusters on layer l_i.

Figure 8.3 illustrates the process for formulating a shared hierarchical framework. The stay points from different users are placed into one dataset, and recursively clustered into several clusters in a divisive manner using a density-based clustering algorithm, such as DBSCAN or OPTICS [7]. As compared to an agglomerative method like K-Means, these density-based approaches are capable of detecting clusters with irregular structures, which may stand for a set of nearby restaurants, a beach, or a shopping street.

As a result, the similar stay points from different users are assigned to the same cluster, and the clusters on different layers denote locations of different granularities. From the top to the bottom of the hierarchy, the geospatial scale of these clusters decreases while the granularity of the locations (corresponding to the clusters) becomes finer. For example, cluster c_{20} on the second layer is divided into two clusters c_{30} and c_{31} on the third layer. So, c_{30} and c_{31} have a smaller size in geographical spaces while they are with a finer granularity than c_{20}. This hierarchical feature is useful for differentiating people with different degrees of similarity. Intuitively, people sharing the same location histories on a deeper layer might be more correlated than on a higher layer. For instance, people visiting the same museum are more likely to be similar than those visiting the same city.

Overall, the shared hierarchical framework provides different users with a uniform foundation to re-formulate their own location history, which looks like a hierarchical graph. Meanwhile, this shared framework is the model representing the location histories of a myriad of users in a LBSN.

8.2.2.3 Constructing Personal Location History

A user's personal hierarchical graph can be constructed by substituting each stay point in the user's original location history (refer to Definition 8.4) with the cluster (from different layers of the shared framework) the stay point pertains to. For example, as illustrated in Fig. 8.3, User 2's location history is originally represented as

$$LocH = s_1 \xrightarrow{\triangle t_1} s_2 \xrightarrow{\triangle t_2} s_3 \xrightarrow{\triangle t_3} s_4 \xrightarrow{\triangle t_4} s_5 \xrightarrow{\triangle t_5} s_6 \xrightarrow{\triangle t_6} s_7 \xrightarrow{\triangle t_7} s_8. \quad (8.4)$$

After projecting these stay points onto the third layer of the shared framework, User 2's location history can be transferred to,

$$LocH = c_{31} \xrightarrow{\triangle t_1} c_{34} \xrightarrow{\triangle t_2} c_{33} \xrightarrow{\triangle t_3} c_{32} \xrightarrow{\triangle t_4} c_{31} \xrightarrow{\triangle t_5} c_{32} \xrightarrow{\triangle t_6} c_{32} \xrightarrow{\triangle t_7} c_{31}. \quad (8.5)$$

Where c_{ij} is the jth cluster on the ith layer. For instance, s_1, s_5, s_6, and s_8 belong to cluster c_{31}. Further, we merge the same cluster (like c_{31}) continuously appearing in a user's location history.

$$LocH = c_{31} \xrightarrow{\triangle t_1} c_{34} \xrightarrow{\triangle t_2} c_{33} \xrightarrow{\triangle t_3} c_{32} \xrightarrow{\triangle t_4} c_{31} \xrightarrow{\triangle t_6} c_{32} \xrightarrow{\triangle t_7} c_{31}. \quad (8.6)$$

This transformation from a stay point to a cluster ID is performed on each layer of the shared framework. As a result, User 2's location history is denoted as a set of sequences of clusters. Since a user could visit a cluster multiple instances at different times, the presentation of a user's location (in sequences) looks more like a hierarchical graph. Generally speaking, the personal hierarchical graph is the integration of two structures: a shared hierarchical framework F and a graph G on each layer of the F. The tree expresses the parent-children (or ascendant-descendant) relationships of the nodes pertaining to different levels, and the graphs specify the peer relationships among the nodes on the same level. Refer to the bottom part of Fig. 8.3 for two examples.

8.2.3 Semantic Model Representing User Location History

In this section, an individual's stay in the physical world is provided with some semantic meanings, e.g., "museum→ cinema → restaurant," aiming to transfer human location history from the geographical spaces into semantic spaces. The semantic meaning of a location reveals the interests of an individual better than its original geo-position, and enables detection of similar users without any overlapping of geographic spaces, e.g., people living in different cities.

Expanding the method introduced in Section 8.2.2, [39] proposed a solution that is comprised of the following three steps: 1) stay point representation in semantic spaces, 2) the formulation of a shared semantic framework, and 3) the construction

of personal location histories. The major difference between this method and that designed for geographical spaces lies in the first step. The three steps are detailed in the following subsections.

8.2.3.1 Stay Point Representation

This step aims to represent a stay point (detected in Section 8.2.2.1) with the semantic meaning (e.g., a restaurant) of the location where the stay occurred. However, it is almost impossible to identify the exact point of interest (POI) an individual has visited given a stay point, because of the GPS positioning error and the crowded distribution of POIs in a city. In practice, as shown in Fig. 8.6, a GPS reading usually has a 10-meter or more error in its real position. Accordingly, there could be multiple POIs of different categories involved in this distance. Unfortunately, the nearest POI to the center of a stay point may not be the actual place that an individual visited. What is worse, many POIs, like restaurants, shopping malls, and cinemas, often overlap in the same building.

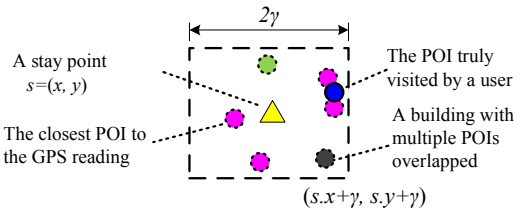

Fig. 8.6 Challenges in discovery of the semantic meaning of a stay point

Due to the challenge mentioned above, it is necessary to first expand a stay point to a stay region covering the POI that a user has visited. For example, as depicted in Fig. 8.6, a stay point s is expanded to a region $[s.x-\gamma, s.x+\gamma] \times [s.y-\gamma, s.y+\gamma]$ where γ is a parameter formulating a bounding box. The value of γ is related to the threshold δ for detecting a stay point.

After that, a feature vector is constructed for each stay region according to the POIs located in a region (defined in Definition 8.6). Here, TF-IDF (term frequency-inverse document frequency) [32, 33], a statistical measurement used to evaluate how important a word is to a document in a collection or corpus, is employed. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Similarly, the method proposed in [39] regards categories of POIs as words and treats stay regions as documents. Intuitively, if POIs of a category occur in a region many times, this POI category is important in representing this region. Furthermore, if a POI category (e.g., "museum" and "natural parks") occurs rarely in other regions, the category is more representative for the region (in which it is located) beyond a common POI category, e.g., "restaurant," which appears in many places. Thus, both

the occurrence frequency of a POI category in a region (similar to TF) and the inverse location frequency (equivalent to IDF) of this category have been considered in [39]. Combining these two factors, the feature vector is defined as follows:

Definition 8.6 (Feature Vector). The feature of a stay region r in a collection of regions R is $f_r = < w_1, w_2 \ldots, w_K >$, where K is the number of unique POI categories in a POI database and w_i is the weight of POI category i in the region r. The value of w_i is calculated as Eq. 8.7:

$$w_i = \frac{n_i}{N} \times \log \frac{|R|}{|\{\text{Regions containing i}\}|} \tag{8.7}$$

Suppose that s_1 contains two restaurants and one museum, and s_2 only has four restaurants. The total number of stay regions created by all the users is 100, in which 50 have restaurants and two contain museums. So, the feature vectors of s_1 and s_2 are f_1 and f_2 respectively:

$$f_1 = (\frac{2}{3} \times \log \frac{100}{50}, \frac{1}{3} \times \log \frac{100}{2}, \ldots),$$
$$f_2 = (\frac{4}{4} \times \log \frac{100}{50}, 0, \ldots).$$

Although we still cannot identify the exact POI category visited by an individual, this feature vector determines the interests of a user to some extent by extracting the semantic meaning of a region accessed by the individual. For example, people are likely to conduct similar activities at similar places. Also, users visiting locations with similar POI categories may have similar interests. Consequently, the representation of a stay point carries advanced semantic information (beyond its geographical position), contributing to a broad range of applications in LBSNs, such as the calculating of similarity between two users in terms of their location histories and activity inferences.

8.2.3.2 Building a Semantic Location History

Step 2: Formulating a shared semantic framework: The second step clusters the stay regions into groups according to their feature vectors. The stay regions in the same cluster can be regarded as locations of similar type with similar semantic meanings. However, a flat clustering is insufficient to differentiate similar users of different extents. Intrinsically, we are more capable of discriminating similar users given categories with a finer granularity. For example, "restaurant" helps identify users who like dining out, while "Indian restaurant" and "Japanese restaurant" enable us to differentiate people interested in different types of food.

Considering this factor, the feature vectors are hierarchically clustered in a divisive manner, building a tree-structured sematic location hierarchy. This is similar to generating a shared framework in the geographical spaces (refer to Section 8.2.2.2). As shown in the middle part of Fig. 8.7, feature vectors of all users are placed into

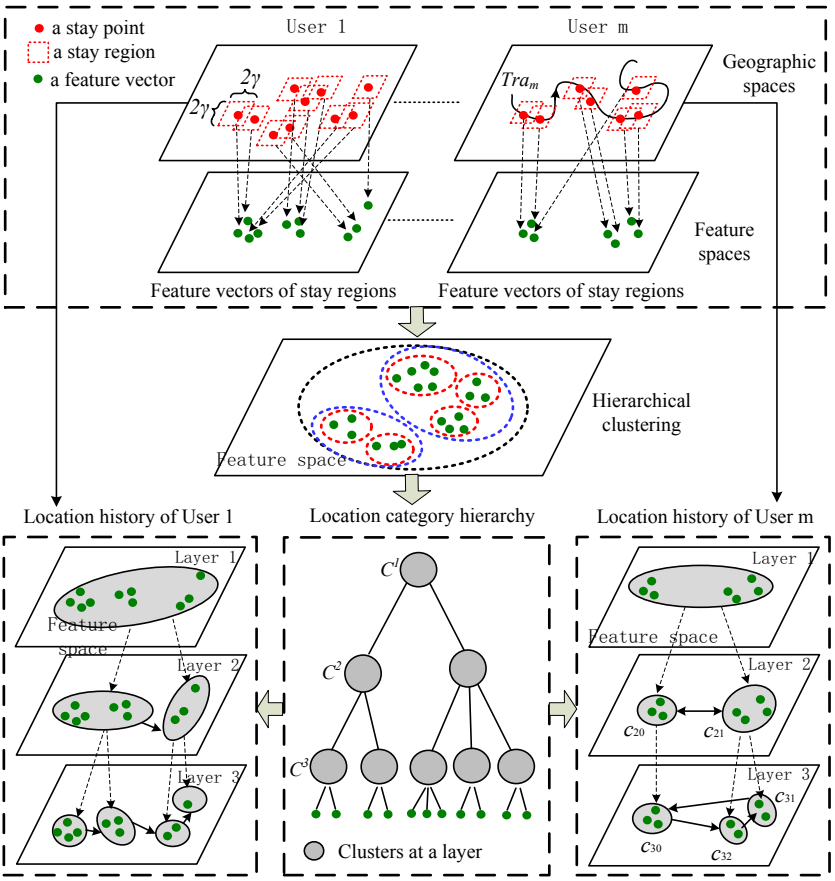

Fig. 8.7 Modeling human location history in semantic spaces

one cluster and this cluster is treated as the root (i.e., cluster at layer 1). Each cluster c at layer j ($j \geq 0$) is split into a set of sub-clusters by using a flat clustering algorithm. The resulting sub-clusters of c are considered c's child nodes at layer $j+1$. This procedure repeats a given number of times, leading to a tree-structured hierarchy where clusters at a lower layer have a finer granularity.

Step 3: Construct personal location history: In the third step, a location history is constructed for each user based on the semantic location hierarchy and the user's stay points. Originally, a user's location history in the geographic spaces is represented by a sequence of stay points with the travel time between each two consecutive stay points. Then, on each layer of the semantic location hierarchy, a stay point is substituted with the semantic location that the stay point's feature vector pertains to. After this projection, different users' location histories become comparable in the semantic spaces.

8.3 Mining User Similarity Based on Location History

8.3.1 Motivation and Overview

As mentioned before, the connection between users in a location-based social network arises from two aspects. One is the original interdependency from an existing social structure, e.g., family, classmates, colleagues, and relatives, or from an online social networking service like Twitter or Facebook. The other is the new interdependency that is derived from the location data generated by the users after they joined a LBSN. The latter is the source of power expanding a location-based social network, essentially differentiating a LBSN from a traditional social network.

The similarity between users' location histories represents the strength of the latter interdependency, thereby determining if a LBSN could expand successfully. This similarity can enable many novel applications, such as friend recommendation and community discovery, in a LBSN. For example, according to this user similarity, a location-based social networking service can recommend to an individual a list of potential friends who might share similar interests with her. The individual can then consider adding these friends to her social structure, or sending a targeted invitation to them when organizing some social activity. Because of the shared interests with the individual, they are more likely to be receptive to such an invitation. Further, a LBSN service can discover new locations (based upon these potential friends' location histories) that match the user's preferences, i.e., a personalized location recommender system.

As discussed previously, a person's location history in the real world implies rich information about their interests and preferences. For example, if a person usually goes to stadiums and gyms, the person might like sports. According to the first law of geography, *everything is related to everything else, but near things are more related than distant things*, people who have similar location histories are more likely to share similar interests and preferences. The more location histories they share, the more correlated these two users would be. Note that the location history mentioned here includes its representation in both geographical and semantic spaces. This claim even makes more sense in the semantic spaces as compared to geographical spaces. That is, people accessing locations with similar semantic meanings like a cinema are more likely to be similar.

[28] is the first publication proposing a framework to estimate the similarity between users in terms of their location histories, followed by a series of similar work [39, 16, 25, 44]. In this framework, the similarity between each pair of users is calculated according to two steps. First, find a set of similar subsequences shared by two users on each layer of their hierarchical graphs. Here a similar sequence stands for two individuals who have visited the same sequence of places for similar time intervals. Second, given the similar sequences, calculate a similarity score for the pair of users involving the following three factors:

- Sequential property of users' movements: This framework takes into account not only the locations they accessed, but also the sequence in which these

locations were visited. The longer the similar sequences shared by two users' location histories are, the more related these two users might be.

- Hierarchical property of geographic spaces: This framework mines user similarity by exploring movements on different scales of geographic (or, semantic) spaces. Users who share similar location histories on a space of finer granularities might be more correlated. For example, people accessing the same building could be more similar then those visiting the same city. In this example, a building belongs to a lower layer of the geographic hierarchy than the city. This claim also holds in semantic spaces. For instance, two users sharing an interest in dining at Chinese restaurants might be more similar than others who generally like dinning in any restaurant. Here, the Chinese restaurant is a subset of restaurants, thereby having a finer granularity.
- Popularity of different locations: Analogous to inverse document frequency (IDF) [34], the proposed framework considers the visited popularity of a location when measuring the similarity between users. Two users who access a location visited by a few people might be more correlated than others who share a location history accessed by many people. For example, a myriad of people have visited the Great Wall, a well-known landmark in Beijing. It might not mean all these people are similar to one another. If two users visited a small museum, however, they might indeed share some similar preferences.

The input of this framework is the location histories (i.e., two hierarchical graphs) of two users in geographical or semantic spaces, and the output is a similarity score indicating how similar these two users are.

8.3.2 Detecting Similar Sequences

In this step, the sub-sequences shared by two users at each layer of their hierarchical graph are determined. Intuitively, users sharing the habit of "cinema→ restaurant→ shopping" are more similar to each other than those visiting these three places separately or in a different order. Therefore, the simple method counting the number of items shared by two sequences will lose a great deal of information about an individual's behavior and preferences. To address this issue, we must consider both the order of visitation and the travel time between two locations when detecting similar sub-sequences. Under the circumstances, *Travel Match* and *Maximum Travel Match* are defined as follows:

Notation: Given sequence $Seq = (c_1 \xrightarrow{\triangle t_1} c_2 \xrightarrow{\triangle t_2} \cdots \xrightarrow{\triangle t_{m-1}} c_m)$, we denote the i-th item of Seq as Seq[i] (e.g., $Seq[1]=c_1$) and represent its subsequence as $Seq[a_1, a_2, \ldots, a_k]$ where $1 \leq a_1 < a_2 < \cdots \leq m$, for instance, $Seq[1,3,6,7]= c_1 \rightarrow c_3 \rightarrow c_6 \rightarrow c_7$.

Definition 8.7 (Travel Match). Given a temporal constraint factor $\rho \in [0,1]$ and two sub-sequences $Seq_1[a_1, a_2, \ldots, a_k]$ and $Seq_2[b_1, b_2, \ldots, b_k]$ from two sequences

Seq_1 and Seq_2 respectively, these two sub-sequences formulate a k-length travel match if they hold the following two conditions:

1. $\forall i \in [1,k]$, $a_i = b_i$, and
2. $\forall i \in [1,k)$, $\frac{|\triangle t_i = \triangle t_i'|}{\max(\triangle t_i, \triangle t_i')} \leq p$, where $\triangle t_i$ is the travel time between a_i and a_{i+1}, and $\triangle t_i'$ denotes that between b_i and b_{i+1}.

This travel match is represented by $(a_1,b_1) \rightarrow (a_2,b_2) \rightarrow \cdots \rightarrow (a_k,b_k)$.

Definition 8.8 (Maximum Travel Match). A travel match $(a_1,b_1) \rightarrow (a_2,b_2) \rightarrow \cdots \rightarrow (a_k,b_k)$ between two sequences Seq_1 and Seq_2 is a maximum travel match if,

1. No left increment: $\nexists a_0 < a_1, b_0 < b_1$, s.t.,
   $(a_0,b_0) \rightarrow (a_1,b_1) \rightarrow (a_2,b_2) \rightarrow \cdots \rightarrow (a_k,b_k)$;
2. No right increment: $\nexists a_{k+1} > a_1, b_{k+1} > b_k$, s.t.,
   $(a_1,b_1) \rightarrow (a_2,b_2) \rightarrow \cdots \rightarrow (a_k,b_k) \rightarrow (a_{k+1},b_{k+1})$;
3. No internal increment: $\forall i \in [1,k], \nexists a_i < a_{i'} < a_{i+1}$ and $b_i < b_{i'} < b_{i+1}$, s.t.,
   $(a_1,b_1) \rightarrow (a_2,b_2) \rightarrow \cdots \rightarrow (a_i,b_i) \rightarrow (a_{i'},b_{i'}) \rightarrow (a_{i+1},b_{i+1}) \rightarrow \cdots \rightarrow (a_k,b_k)$

Essentially, a travel match is a common sequence of locations visited by two users in a similar amount of time, and a maximum travel match is a travel match that is not contained in any other travel matches. Note that 1) the locations in a travel match do not have to be consecutive in the user's original location history, and 2) what we need to detect for the calculating of user similarity are the maximum travel matches. Additionally, the location in a travel match can be a cluster of stay points in the geographical spaces, or a cluster in semantic spaces.

Figure 8.8 demonstrates an example of a maximum travel match between two sequences Seq_1 and Seq_2. Here, a node stands for a location and the letter in a node represents the ID of the location. The numbers on the top of the box denotes the index of a node in a sequence, e.g., location A is the first node in both Seq_1 and Seq_2. The number appearing on a solid edge means the travel time between two consecutive nodes, and the number shown on a dashed edge denotes the duration that a user stayed in a location.

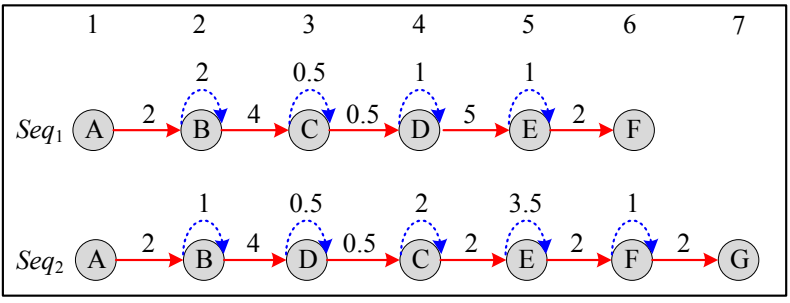

Fig. 8.8 An example of finding maximal travel match

Let $\rho = 0.2$ in this example. First, $(1,1) \rightarrow (2,2)$, i.e., $A \rightarrow B$, is a travel match, because the travel times ($A \rightarrow B$) in Seq_1 and Seq_2 are identical, $|2-2|/2 = 0$. Then, we find that $(2,2) \rightarrow (3,4)$, i.e., $B \rightarrow C$, also satisfies the conditions defined in Definition 8.7. Though B and C are not directly connected in Seq_2, the travel time between these two locations is $4+0.5+0.5 = 5$, which is very similar to that of Seq_1. In short, $|5-4|/5 = 0.2$. However, both $A \rightarrow B$ and $B \rightarrow C$ are not the maximum travel match in this example as they are contained in $A \rightarrow B \rightarrow C$, i.e., $(1,1) \rightarrow (2,2) \rightarrow (3,4)$. Later, $C \rightarrow E$ and $C \rightarrow F$ cannot formulate travel matches due to the difference between corresponding travel times. Using the same approach, we find $(1,1) \rightarrow (2,2) \rightarrow (4,3) \rightarrow (5,5) \rightarrow (6,6)$, i.e., $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F$, is another maximum travel match. Overall, we detect two maximum travel matches, $A \rightarrow B \rightarrow C$ and $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F$ from Seq_1 and Seq_2.

Some well-known sequence matching algorithms, such as longest common subsequences (LCSS) searching [36]] and dynamic time wrapping (DTW) [43], cannot satisfy the need to discover the maximum travel matches as they do not incorporate the travel time between two locations in the matching process. Due to this reason, a method has been proposed in [39] for detecting the maximum travel matches from two sequences. This method consists of two steps, summarized as follows:

The first step detects the 1-length travel matches between two sequences and identifies a precedence relation between these 1-length matches. For example, A in Fig. 8.8, i.e., (1,1), is a 1-length travel match between Seq_1 and Seq_2, and A is a precedence of B. Then, the 1-length matches and their precedence relation are transferred into a precedence graph G, where a node is a 1-length match and an edge corresponds to the precedence relation between 1-length matches.

The second step searches graph G for the maximum length path which has been proved equivalent to the maximum matches.

Following the case illustrated in Fig. 8.8, Fig. 8.9 shows an example of building graph G based on Seq_1 and Seq_2. As demonstrated in Fig. 8.9 A), the identical items in two sequences are first detected by putting these two sequences into a matching matrix. The numbers that stand on the top and left of the matrix denote the index of an item in a sequence. For example, A_{11} means that A is the first item in both sequences. In Fig. 8.9 B), each node corresponds to a trivial match, and an edge between two nodes stands for a precedent relation between two trivial matches. The number in a node indicates its order being added to the graph. For instance, F_{66} is the first node being added to graph G.

After the graph building process, precedence graph G is a directed acyclic graph in which a path represents a travel match (between two sequences). More specifically, if $(a_1,b_1) \rightarrow (a_2,b_2) \rightarrow \cdots \rightarrow (a_k,b_k)$ is a path in G, $Seq_1[a_1,a_2,\ldots,a_k]$ and $Seq_2[b_1,b_2,\ldots,b_k]$ form a travel match, and vice versa. Meanwhile, path P in G corresponds to a maximum travel match if the first node of P has zero in-degree and the last node has zero out-degree. For instance, path $A_{11} \rightarrow B_{22} \rightarrow C_{34}$ in Fig. 8.9 b) corresponds to the maximum travel match $(1,1) \rightarrow (2,2) \rightarrow (3,4)$ in Fig. 8.8.

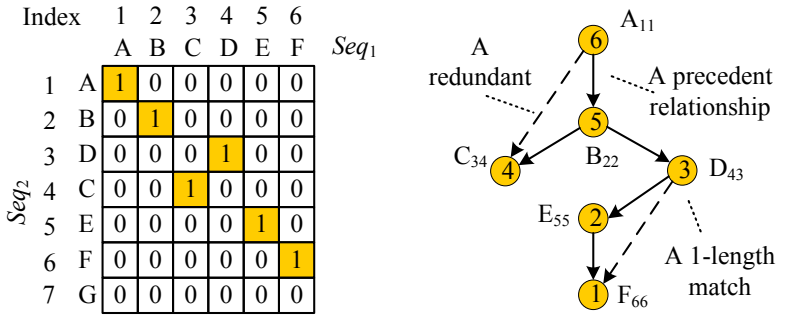

Fig. 8.9 The precedence graph for Seq_1 and Seq_2

8.3.3 Calculating Similarity Scores

After detecting the maximum travel matches from two users' location sequences, a similarity score can be calculated for the two users according to the following three factors: visited popularity of a location, sequential properties, and hierarchical properties, which were introduced in the beginning of Section 8.3.2 and are formally defined in Eq. 8.8, 8.9, 8.10, and 8.11.

$$SimUser(LocH_1, LocH_2) = \sum_{l=1}^{L} f_w(l) \times SimSq(Seq_1^l, Seq_2^l); \tag{8.8}$$

$$SimSq(Seq_1, Seq_2) = \frac{\sum_{j=1}^{m} simTM(t_j)}{|Seq_1| \times |Seq_2|}, \tag{8.9}$$

$$SimTM(s) = g_w(k) \times \sum_{i=1}^{k} vp(c_i); \tag{8.10}$$

$$vp(c) = \log \frac{N}{n}, \tag{8.11}$$

where N is the total number of users in the dataset and n is the number of users visiting location c.

Given two users' location histories $LocH_1$ and $LocH_2$, the similarity between them can be computed by summing up the similarity score at each layer of the hierarchical graph (refer to Definition 8.5 for details) in a weighted way. A function $f_w(l)$ is employed to assign a bigger weight to the similarity of sequences occurring at a lower layer, e.g., $f_w(l) = 2^{l-1}$, where l is the depth of a layer in the hierarchy.

Then, the similarity between two sequences Seq_1 and Seq_2 at a layer, $SimSq(Seq_1, Seq_2)$, is represented by the sum of the similarity score, $simTM(t_j)$, of each maximum travel match between Seq_1 and Seq_2. Here, m is the total number of maximum matches. Meanwhile, $SimSq(Seq_1, Seq_2)$ is normalized by the production of

the lengths of the two sequences, since a longer sequence has a higher probability of having long matches. That is, a user with a longer history of data is more likely to be similar to others (than a user having a shorter period of data) without performing the normalization.

Further, the similarity score of a maximum travel match t, $simTM(t)$, is calculated by summing up the vp (visited popularity) of each location c contained in t. At the same time, the $simTM(t)$ is weighted in terms of the length k of t, e.g., $g_w(k) = 2^{k-1}$. The insight leading to Eq. 8.10 and 8.11 is based on two aspects. First, the longer the similar sequences shared by two users' location histories, the more related these two users are likely to be (this is known as the sequential property). Second, users who have accessed a location visited by a few people might be more correlated than others who share a location history accessed by many people (the visited popularity of a location). According to the experimental results, it was discovered that the number of shared sub-sequences exponentially decreases with the increase of the length of the sub-sequence. So, in the implementation, it's preferable to use an exponential weight function, assigning a higher weight to the longer sequences.

Note that this framework can be applied to the location history modeled either in geographical or semantic spaces. Specifically, when applying this framework to the location history in geographical spaces, a location is a cluster of stay points as depicted in Fig. 8.3, while a location is replaced by a group of semantic features in the semantic spaces illustrated in Fig. 8.7.

8.4 Friend Recommendation and Community Discovery

8.4.1 Methodology

With the user similarity calculated above, we can hierarchically cluster users into groups in a divisive manner by using some clustering algorithms like K-mean. Consequently, as depicted in Fig. 8.10, we can build a user cluster hierarchy, where a cluster denotes a group of users sharing some similar interests and different layers represent different levels of similarity. The clusters shown on a higher layer could stand for big communities in which people share some high-level interests, such as sports. The clusters occurring on the lower layers denote people sharing some narrower interests, like hiking (the layer of the hierarchy can be determined based on the needs of applications). Meanwhile, we can find one representative user (the center) for each cluster according to the similarity scores between each pair of users. For instance, the individual with the minimal distance to other users in the cluster (the individual pertains to) can be selected as the representative user of the cluster.

This user hierarch brings us two types of advantages:

1) Fast retrieval of similar users: Instead of checking all the users, we can retrieve the top k similar users for an individual by only ranking the users from the same cluster (the individual belongs to) in terms of similarity score. This retrieval process

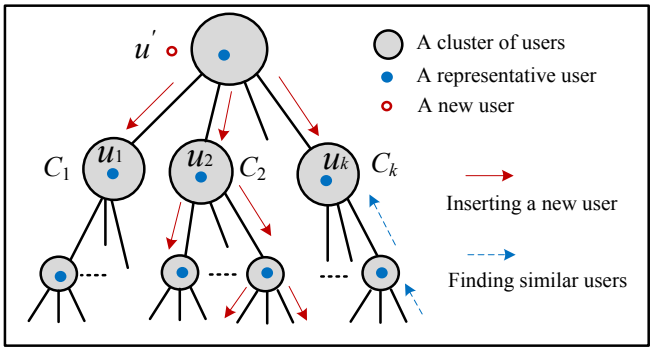

Fig. 8.10 Finding similar users and inserting new users in hierarchical user clusters

can start from the bottom layer of the hierarchy, as depicted by the blue dash arrow in Fig. 8.10. If the number of users is less than k in the bottom-layer cluster, we can further check the parent node (cluster) of this cluster until finding a cluster with more than k users.

2) Insert new users: When a new user u' enters the system, it is not necessary to compute the similarity score between u' and each user in the system. This process is very time consuming and will become more difficult as the number of users increases. Instead, we only need to insert this user into the most appropriate clusters on each layer of the hierarchy by computing the similarity between u' and the representative user in a cluster. For example, as demonstrated by the red solid arrows in Fig. 8.10, we first compute the similarity between u' and $(u_1, u_2, \ldots, u_k)$ who are representative users in each cluster. If u_2 is the most similar user to u' out of the k users, we insert u' into u_2's cluster C_2. Then, we further check the children clusters of C_2 and insert u' into the clusters whose representative user is the most similar to u'. This process is performed iteratively until reaching the bottom layer of the hierarchy.

In practice, we do not need to re-build this hierarchy unless the number of newly inserted users exceeds a certain threshold. That is, in most cases we can find similar users for a person very efficiently.

Evaluating the applications in a location-based social network, such as friend recommendation and community discovery, is a non-trivial research topic due to the following challenges: data, ground truth, and metrics.

8.4.2 Public Datasets for the Evaluation

The biggest challenge of the evaluation comes from the data, consisting of location data such as GPS trajectories and the social structure, of many users. To collect the data, a research group typically needs to deploy a location-based social network-

ing service and encourage enough people to use this service in a certain period, e.g., 3 months. Without an online LBSN service, they could assign some location-acquisition devices like GPS loggers to a group of users and collect the data offline. Both ways are very time-consuming and resource-intensive, thereby becoming a major barrier to many professionals stepping into this field.

In recent years, a few real-world datasets created by some pioneers were made available on the Internet for free download, for example, "the reality mining dataset" [5] from MIT media laboratory and "GeoLife GPS Trajectories" [4] from Microsoft research. The reality mining dataset was collected by one hundred human subjects with a Bluetooth-enabled mobile phone over the course of nine months, representing 500,000 hours of data on users' location, communication, and device usage behavior.

The GeoLife GPS Trajectories was collected by 170 users with a GPS logger or GPS-phone (see Fig 8.11) over a period of four years (from April 2007 to the date when this book was published). This dataset is still growing and upgrading with an annual release. The latest version (released in July, 2011) is comprised of 17,085 GPS trajectories with a total distance over 1,000,000km and an effective duration over 48,000 hours. 95 percent of these trajectories are logged in a dense representation, e.g., every 2∼5 seconds or every 5∼10 meters per point. Figure 8.12 shows the distribution of this dataset in the urban area of Beijing, where the figures associated with the colored bar indicate the number of GPS points in a location.

Fig. 8.11 GPS devices used for collecting data in GeoLife Project

This dataset recorded a broad range of users' outdoor movements, including not only daily routines like going to work but also some entertainment and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. A part of these trajectories has a label of transportation modes including driving, riding a bike, taking a bus, and walking. These datasets provide professionals with a good resource to evaluate their early research into LBSN, significantly boosting the LBSN community. Detailed information can be found at the website [4].

The advent of some commercial LBSN services like Foursquare brings new opportunities to carry out evaluations using large-scale and real-world data. For example, Foursquare released an API set allowing LBSN researchers to crawl the publically available check-in records generated by users. The collected data includes the venue where a user checked in and a corresponding timestamp as well as the tips

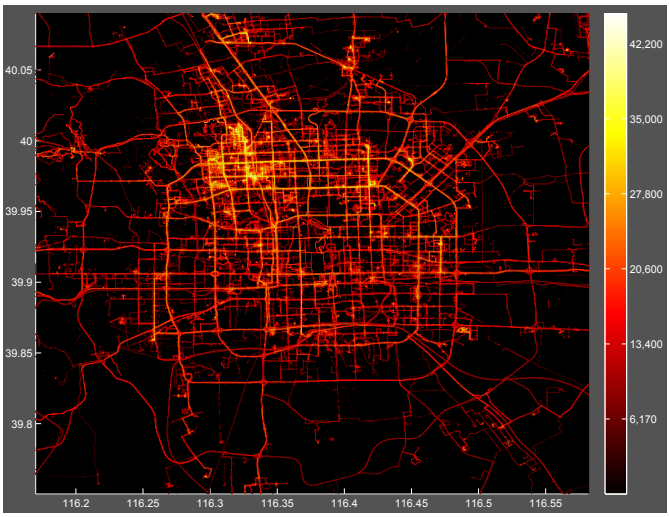

Fig. 8.12 The distribution of GeoLife dataset in the urban area of Beijing

that the user left in the venue. At the same time, the social structure of a user can be obtained by using this API. A great deal of research based on such data has been published [16, 41, 40], verifying some hypothesis proposed in LBSN, e.g., people with similar location histories can be correlated.

8.4.3 Methods for Obtaining Ground Truth

The second challenge stems from ground truth. For example, to evaluate a friend recommendation, we need to rank people according to the similarity inferred in terms of location histories. The ability to obtain an idea ranking (i.e., ground truth) is important. Generally speaking, there are two ways of generating ground truth. One is performing a questionnaire-style user study. The other is to extract ground truth from an individual's social structure [17, 38, 9, 31].

The former approach usually provides the users who collect location data for the research with a questionnaire inquiring about their interests. In the GeoLife project, for example, each user answered the questions shown in Fig. 8.13 A) by giving a rank ($1\sim4$) to denote different degrees of desire for an activity. A user's answer, e.g., Fig. 8.13 B), is regarded as an interest vector, in which each entry is the user's rank to a corresponding question. In the example, the user's interest vector is $< 3,2,1,4,3,1,3,1,2,3,1,1 >$. A cosine similarity between two users' interest vectors can be calculated and used to rank a group of people for an individual. As a result, the top k people can be retrieved as a ground truth. Due to the intensive human effort, this kind of approach can only be applied to a small scale of people such as when using the Reality Mining Dataset and GeoLife GPS Trajectories.

Where do you like to go in weekends? Please rank from 1(dislike) to 4(favorite).	Example response
1. Shopping	3
2. Theatre	2
3. Karaoke	1
4. Go out for dinner	4
5. Outdoor sports, e.g., hiking	3
6. Indoor sports, e.g., gym and bowling	1
7. Natural parks	3
8. Exhibition, museum	1
9. Stay home; not go to any places	2
10. Go to office; over-time working	3
11. Visit parents, relatives, or friends	1
12. Campus	1
A)	B)

Fig. 8.13 A questionnaire A) and an example of answers B)

The latter approach uses the closeness between two users inferred from the connections in their social structure as the ground truth. For example, random walk theory [17] can be used to analyze the closeness of two nodes (i.e., friendship strength in a social network context) using the resistance distance, which is the random walk steps for the electrons traveling from one node to the other, in a social graph. However, the random walk theory completely overlooks semantic information contained in social networks. As a result, recently, more advanced research have been proposed to analyze the closeness between two users considering: 1) the similarity of their profile [38, 11], e.g., demographics like age, gender, and hometown, and 2) interaction activities [9, 22, 31], e.g., commenting, tagging, and group communication patterns.

8.4.4 Metrics for the Evaluation

The third challenge is the metric used to measure the effectiveness of inferred user similarity, given the data and ground truth. As mentioned before, user similarity is a metric specifying to what extent two users are similar to each other, instead of a binary value indicating whether two users are similar or not. The goal is to rank a group of people for an individual according to similarity and recommend the top k people as potential friends to the individual. Accordingly, it is natural to look at user similarity as an information retrieval problem.

Given an individual, the top k similar users to the individual can be retrieved according to their similarity scores (inferred by the approach mentioned previously). An idea rank can be formulated from the ground truth (which was obtained by using one of the methods mentioned in Section 8.4.3). Based on these two ranking lists, *MAP* (Mean Average Precision) and *nDCG* (Normalized Discounted Cumulated

Gain) are calculated for retrieval. After testing all users, a mean value of *MAP* and *nDCG* is computed respectively.

More specifically, when generating the ground truth for an individual, users are divided into groups according to their similarity scores to the individual. As demonstrated in Fig. 8.14, users are ranked in terms of the similarity scores to the individual, and then split into 5 classes: 0∼4. The users in class 4 have a higher similarity score than those in a lower class. The split can be driven by evenly partitioning the similarity scores or by a uniform division of users. The number of classes is determined by application which can assign each cluster a semantic meaning as in the example shown in Fig. 8.14. Afterwards, the numeric value of a (ground truth) similarity is replaced by the class ID it pertains to.

In the testing phase, the top k users are retrieved for an individual according to our method (based on location history). Then, a ranking list e.g., $G = (U_3, U_2, \ldots, U_5)$ can be obtained. By replacing these user IDs with the corresponding class IDs, another ranking list, e.g., $G = (4,3,2,3,,0)$, is formulated. Now, a score for this ranking list can be calculated in terms of *nDCG*. *nDCG* is used to compute the relative-

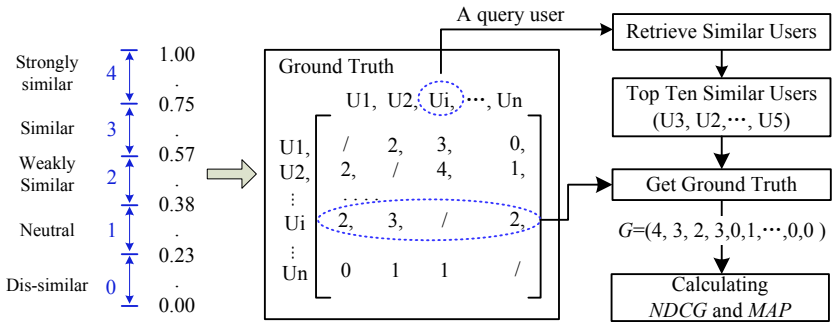

Fig. 8.14 Evaluation metric for user similarity detection

to-the-ideal performance of information retrieval techniques [26]. The discounted cumulative gain of G is computed as follows: (In our experiments, $b = 2$.)

$$DCG[i] = \begin{cases} G[1], & \text{if } i = 1 \\ DCG[i-1] + G[i], & \text{if } i < b \\ DCG[i-1] + \frac{G[i]}{\log_b i}, & \text{if } i \geq b \end{cases} \tag{8.12}$$

Given the ideal discounted cumulative gain DCG', then *nDCG* at i-th position can be computed as $nDCG[i] = DCG[i]/DCG'[i]$. According to Eq. 8.12, $nDCG[3]$ of $G = (4,2,3,3,0)$ can be calculated as follows:

$$DCG[1] = G[1] = 4;$$
$$DCG[2] = DCG[1] + G[2] = 4 + 2 = 6;$$
$$DCG[3] = DCG[2] + (G[3])/(\log_2 3) = 6 + 1.893 = 7.893;$$

However, the idea ranking should be $G' = (4,3,3,2,0)$. According to the same method, the $DCG'[3] = 8.893$. As a result,

$$nDCG[3] = \frac{DCG[3]}{DCG'[3]} = \frac{7.893}{8.893} = 0.888.$$

8.5 Summary

This chapter defined a location-based social network and discussed a research philosophy from the perspective of user and location. Three categories of location-based social networking services were classified in terms of the location data powering a service and a user's preferences in the service. Then, research focusing on understanding users in a location-based social network was gradually explored from modeling the location history of an individual to estimating the similarity between different users, and then moving to high-level applications, such as friend recommendation and community discovery. Some possible methods for evaluation of these applications were discussed, and a number of publically available datasets have been listed as well. All these efforts are enabled by the unprecedented wealth of user-generated trajectories.

References

1. Bikely. http://www.bikely.com
2. Flickr. http://www.flickr.com
3. Foursquare. https://foursquare.com
4. GeoLife GPS Trajectories. http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/default.aspx
5. The Reality Mining Dataset. http://reality.media.mit.edu/dataset.php
6. Twitter. http://twitter.com
7. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. In: Proceedings of the 1999 ACM SIGMOD international conference on Management of data, SIGMOD '99, pp. 49–60. ACM, New York, NY, USA (1999)
8. Ashbrook, D., Starner, T.: Using gps to learn significant locations and predict movement across multiple users. Personal Ubiquitous Comput. **7**, 275–286 (2003)
9. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11, pp. 635–644. ACM, New York, NY, USA (2011)
10. Cao, X., Cong, G., Jensen, C.S.: Mining significant semantic locations from gps data. Proc. VLDB Endow. **3**, 1009–1020 (2010)
11. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make new friends, but keep the old: recommending people on social networking sites. In: Proceedings of the 27th international conference on Human factors in computing systems, CHI '09, pp. 201–210. ACM, New York, NY, USA (2009)

12. Chen, Y., Jiang, K., Zheng, Y., Li, C., Yu, N.: Trajectory simplification method for location-based social networking services. In: Proceedings of the 2009 International Workshop on Location Based Social Networks, LBSN '09, pp. 33–40. ACM, New York, NY, USA (2009)
13. Chen, Z., Shen, H.T., Zhou, X., Zheng, Y., Xie, X.: Searching trajectories by locations: an efficiency study. In: Proceedings of the 2010 international conference on Management of data, SIGMOD '10, pp. 255–266. ACM, New York, NY, USA (2010)
14. Chow, C.Y., Bao, J., Mokbel, M.F.: Towards location-based social networking services. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, pp. 31–38. ACM, New York, NY, USA (2010)
15. Counts, S., Smith, M.: Where were we: communities for sharing space-time trails. In: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems, GIS '07, pp. 10:1–10:8. ACM, New York, NY, USA (2007)
16. Cranshaw, J., Toch, E., Hong, J., Kittur, A., Sadeh, N.: Bridging the gap between physical location and online social networks. In: Proceedings of the 12th ACM international conference on Ubiquitous computing, Ubicomp '10, pp. 119–128. ACM, New York, NY, USA (2010)
17. Doyle, P.G., Snell, J.L.: Random walks and electric networks (1984)
18. Doytsher, Y., Galon, B., Kanza, Y.: Querying geo-social data by bridging spatial networks and social networks. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, pp. 39–46. ACM, New York, NY, USA (2010)
19. Eagle, N., de Montjoye, Y.A., Bettencourt, L.M.A.: Community computing: Comparisons between rural and urban societies using mobile phone data. In: Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04, pp. 144–150. IEEE Computer Society, Washington, DC, USA (2009)
20. Eagle, N., Pentland, A., Lazer, D.: Inferring social network structure using mobile phone data. Proceedings of the National Academy of Sciences (PNAS) **106**, 15,274–15,278 (2007)
21. Eagle, N., (Sandy) Pentland, A.: Reality mining: sensing complex social systems. Personal Ubiquitous Comput. **10**, 255–268 (2006)
22. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: Proceedings of the 27th international conference on Human factors in computing systems, CHI '09, pp. 211–220. ACM, New York, NY, USA (2009)
23. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**, 61–70 (1992)
24. Hariharan, R., Toyama, K.: Project lachesis: Parsing and modeling location histories. In: Proceedings of the 3rd International Conference on Geographic Information Science, pp. 106–124 (2004)
25. Hung, C.C., Chang, C.W., Peng, W.C.: Mining trajectory profiles for discovering user communities. In: Proceedings of the 2009 International Workshop on Location Based Social Networks, LBSN '09, pp. 1–8. ACM, New York, NY, USA (2009)
26. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. **20**, 422–446 (2002)
27. Lee, R., Sumiya, K.: Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, pp. 1–10. ACM, New York, NY, USA (2010)
28. Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., Ma, W.Y.: Mining user similarity based on location history. In: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems, GIS '08, pp. 34:1–34:10. ACM, New York, NY, USA (2008)
29. Liu, W., Zheng, Y., Chawla, S., Yuan, J., Xie, X.: Discovering spatio-temporal causal interactions in traffic data streams. In: The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD '11. ACM, New York, NY, USA (2011)
30. Nakamura, A., Abe, N.: Collaborative filtering using weighted majority prediction algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, pp. 395–403. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)

31. Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., Leiser, N., Matias, Y., Merom, R.: Suggesting friends using the implicit social graph. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '10, pp. 233–242. ACM, New York, NY, USA (2010)
32. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**, 513–523 (1988)
33. Salton, G., Fox, E.A., Wu, H.: Extended boolean information retrieval. Commun. ACM **26**, 1022–1036 (1983)
34. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval, pp. 132–142. Taylor Graham Publishing, London, UK, UK (1988)
35. Tang, L.A., Zheng, Y., Xie, X., Yuan, J., Yu, X., Han, J.: Retrieving k-nearest neighboring trajectories by a set of point locations. In: The 12th Symposium on Spatial and Temporal Databases (2011)
36. Vlachos, M., Gunopoulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: Proceedings of the 18th International Conference on Data Engineering, ICDE '02, pp. 673–684. IEEE Computer Society, Washington, DC, USA (2002)
37. Wang, L., Zheng, Y., Xie, X., Ma, W.Y.: A flexible spatio-temporal indexing scheme for large-scale gps track retrieval. In: Proceedings of the The Ninth International Conference on Mobile Data Management, pp. 1–8. IEEE Computer Society, Washington, DC, USA (2008)
38. Xiang, R., Neville, J., Rogati, M.: Modeling relationship strength in online social networks. In: Proceedings of the 19th international conference on World wide web, WWW '10, pp. 981–990. ACM, New York, NY, USA (2010)
39. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Finding similar users using category-based location history. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, pp. 442–445. ACM, New York, NY, USA (2010)
40. Ye, M., Shou, D., Lee, W.C., Yin, P., Janowicz, K.: On the semantic annotation of places in location-based social networks. In: The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD '11. ACM, New York, NY, USA (2011)
41. Ye, M., Yin, P., Lee, D.L., Lee, W.C.: Exploiting geographical influence for collaborative point-of-interests recommendation. In: The 34th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '11. ACM, New York, NY, USA (2011)
42. Ye, Y., Zheng, Y., Chen, Y., Feng, J., Xie, X.: Mining individual life pattern based on location history. In: Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, MDM '09, pp. 1–10. IEEE Computer Society, Washington, DC, USA (2009)
43. Yi, B.K., Jagadish, H.V., Faloutsos, C.: Efficient retrieval of similar time sequences under time warping. In: Proceedings of the Fourteenth International Conference on Data Engineering, ICDE '98, pp. 201–208. IEEE Computer Society, Washington, DC, USA (1998)
44. Ying, J.J.C., Lu, E.H.C., Lee, W.C., Weng, T.C., Tseng, V.S.: Mining user similarity from semantic trajectories. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, pp. 19–26. ACM, New York, NY, USA (2010)
45. Yoon, H., Zheng, Y., Xie, X., Woo, W.: Smart itinerary based on user-generated gps trajectories. In: Proceedings of the 7th international conference on Ubiquitous intelligence and computing, UIC'10, pp. 19–34. Springer-Verlag, Berlin, Heidelberg (2010)
46. Yoon, H., Zheng, Y., Xie, X., Woo, W.: Social itinerary recommendation from user-generated digital trails. Personal and Ubiquitous Computing (2011)
47. Yuan, J., Zheng, Y., Xie, X., Sun, G.: Driving with knowledge from the physical world. In: The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD '11. ACM, New York, NY, USA (2011)
48. Yuan, J., Zheng, Y., Xie, X., Sun, G.: Where to find the next passenger. In: Proceedings of the 13th ACM international conference on Ubiquitous computing, Ubicomp '11. ACM, New York, NY, USA (2011)

49. Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: driving directions based on taxi trajectories. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, pp. 99–108. ACM, New York, NY, USA (2010)
50. Zheng, V.W., Cao, B., Zheng, Y., Xie, X., Yang, Q.: Collaborative filtering meets mobile recommendation: A user-centered approach. In: Proceedings of AAAI conference on Artificial Intelligence (AAAI 2010), pp. 236–241. ACM, New York, NY, USA (2010)
51. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with gps history data. In: Proceedings of the 19th international conference on World wide web, WWW '10, pp. 1029–1038. ACM, New York, NY, USA (2010)
52. Zheng, Y., Chen, Y., Li, Q., Xie, X., Ma, W.Y.: Understanding transportation modes based on gps data for web applications. ACM Trans. Web **4**, 1:1–1:36 (2010)
53. Zheng, Y., Chen, Y., Xie, X., Ma, W.Y.: Geolife2.0: A location-based social networking service. In: Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, MDM '09, pp. 357–358. IEEE Computer Society (2009)
54. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y.: Understanding mobility based on gps data. In: Proceedings of the 10th international conference on Ubiquitous computing, UbiComp '08, pp. 312–321. ACM, New York, NY, USA (2008)
55. Zheng, Y., Liu, L., Wang, L., Xie, X.: Learning transportation mode from raw gps data for geographic applications on the web. In: Proceeding of the 17th international conference on World Wide Web, WWW '08, pp. 247–256. ACM, New York, NY, USA (2008)
56. Zheng, Y., Liu, Y., Xie, X.: Urban computing with taxicabs. In: Proceedings of the 13th ACM international conference on Ubiquitous computing, Ubicomp '11. ACM, New York, NY, USA (2011)
57. Zheng, Y., Wang, L., Zhang, R., Xie, X., Ma, W.Y.: Geolife: Managing and understanding your past life over maps. In: Proceedings of the The Ninth International Conference on Mobile Data Management, pp. 211–212. IEEE Computer Society, Washington, DC, USA (2008)
58. Zheng, Y., Xie, X.: Learning location correlation from gps trajectories. In: Proceedings of the 2010 Eleventh International Conference on Mobile Data Management, MDM '10, pp. 27–32. IEEE Computer Society, Washington, DC, USA (2010)
59. Zheng, Y., Xie, X.: Learning travel recommendations from user-generated gps traces. ACM Trans. Intell. Syst. Technol. **2**, 2:1–2:29 (2011)
60. Zheng, Y., Xie, X., Ma, W.Y.: Geolife: A collaborative social networking service among user, location and trajectory. IEEE Data Eng. Bull. **33**(2), 32–39 (2010)
61. Zheng, Y., Xie, X., Zhang, R., Ma, W.Y.: Searching your life on web maps. In: SIGIR Workshop on Mobile Information Retrieval (2008)
62. Zheng, Y., Yuan, J., Xie, W., Xie, X., Sun, G.: Drive smartly as a taxi driver. In: Proceedings of the 2010 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, UIC-ATC '10, pp. 484–486. IEEE Computer Society, Washington, DC, USA (2010)
63. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.Y.: Recommending friends and locations based on individual location history. ACM Trans. Web **5**, 5:1–5:44 (2011)
64. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining correlation between locations using human location history. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09, pp. 472–475. ACM, New York, NY, USA (2009)
65. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from gps trajectories. In: Proceedings of the 18th international conference on World wide web, WWW '09, pp. 791–800. ACM, New York, NY, USA (2009)