A Travel Route Recommendation System Based on Smart Phones and IoT Environment

Chenzhong Bin, Tianlong Gu, Yanpeng Sun, Liang Chang, and Lei Sun

1 School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China
2 Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
3 Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin 541004, China
4 School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin 541004, China
5 School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Chenzhong Bin; binchenzhong@163.com

Received 15 March 2019; Revised 26 May 2019; Accepted 1 July 2019; Published 14 July 2019

Tourism recommendation systems play a vital role in providing useful travel information to tourists. However, existing systems rarely aim at recommending tangible itineraries for tourists within a specific POI due to their lack of onsite travel behavioral data and related route mining algorithms. To this end, a novel travel route recommendation system is proposed, which collects tourist onsite travel behavior data automatically regarding a specific POI based on smart phone and IoT technology. Then, the proposed system preprocesses the behavior data to transform raw behavior sequences into Tourist-Behavior pattern sequences. Subsequently, the system discovers frequent travel routes from the generated pattern sequences by using an original route mining algorithm, named Tourist-Behavior PrefixSpan. Finally, a route-recommending method is designed to search and rank tangible travel routes according to the querying tourist’s profile and constraint. The experimental results demonstrate that the proposed system is efficient and effective in recommending POI-oriented tangible travel routes considering tourists’ route constraints and personal profile while ensuring that the suggested routes have considerable route values.

1. Introduction

Tourism is a popular leisure activity with the goal of visiting some Points of Interests (abbr. POIs) based on one's personal preference and constraints. Recently, industry and academia have been studying and developing tourism recommendation systems for providing convenient travel information to tourists, including next POI suggestion [1–7], Top-k POIs recommendation [8–10], and POIs travel route recommendation [11–18]. Particularly, travel route recommendations are more practical and useful than the two former kinds of POI recommendations in practice, yet they are more challenging. Travel route recommendation aims to organize a bundle of candidate POIs as a reasonable visit sequence (i.e., an itinerary) while adhering to personal constraints of a given tourist, for example, a limited time or finance budget, user-specified start, and end locations.

Thus, many researchers have studied travel route recommendation issues and designed various algorithms for solving these problems [19]. Most of these works are orienting to city-level or district-level itinerary recommendation scenario, that is, planning a POIs travel route within a city or region for tourists. On the other side, it is difficult for tourists to instantly choose personal interested exhibits or spots and arrange these items into an itinerary under their time budget when travelling in a never visit POI, for example, a museum or a park. This situation makes most of tourists roaming or missing some potential interested items in the POI. However, few existing systems manage to recommend tangible itineraries for tourists within a given POI due to lack of rich onsite travel behavior data and related itinerary mining algorithms.

The popularity of smart phones and the flourish Internet of Things (IoT) techniques provide various means to sense onsite travel behaviors of tourists, including not only travel spatial-temporal trajectories [20] and visit durations but also the tangible travel behaviors [21] such as taking photos, standing, or walking. It is a common sense that one's onsite
travel behaviors imply his or her objective preferences and interests to some objects. For instance, tourists will spend longer visit duration or take more pictures or stand still more times to appreciate something on a spot if they are more interested in something. Thus, gathering tourists’ onsite travel behaviors and mining their personal preferences and frequent travel routes could be an effective approach for recommending tangible travel routes for new similar tourists in specific POIs.

To that end, in this work, we proposed a POI-oriented travel route recommendation system based on IoT technology and smart phones. In detail, we first adopted Bluetooth low energy (BLE) beacons [22] to periodically broadcast positioning information for nearby smart phones. Also, we developed a client App running on an Android smartphone to collect onsite travel behaviors data and corresponding personal profiles and then upload collected data to the system server. Next, on the system server side, all collected travel behavior data are classified according to their personal profiles. And then a behavior sequence preprocessing method and Tourist-Behavior pattern mining algorithms were designed to generate diverse tangible travel routes. At route recommendation stage, to ensure the personalization of our recommendations, the proposed route ranking method recommends tangible travel routes for new tourists by using their personal profiles and route constraints. As all tangible travel routes are constructed from real historical onsite travel behavior, the recommended routes have high accuracy and rationality in terms of visit arrangements. Also, since the recommended routes are retrieved from the corresponding candidate route subset according to the querying tourist’s profile, the visit objects of the final route can suit the personal interests of the tourist better.

Our main contributions are summarized as follows. (a) An onsite travel behavior data collecting method, which is based on tourists’ smartphones and Bluetooth low energy (BLE) beacons, is designed to automatically sense onsite travel behavior under indoor and outdoor tourism scenarios. (b) Tourist-Behavior PrefixSpan algorithm is proposed to generate diverse frequent travel routes effectively based on historical Tourist-Behavior pattern sequences. (c) Travel route ranking method is proposed to recommend a list of tangible travel routes according to the querying tourist’s profile and constraints so as to ensure the route value and rationality of the final travel routes. (d) Experimental results demonstrate the effectiveness of our system in recommending personalized tangible travel routes for tourists in a given POI based on historical onsite travel behavior.

The rest of the paper is organized as follows. Section 2 discusses the related works regarding travel route recommendation systems and tourism recommendation within IoT environment. Section 3 presents the research methodologies of the proposed system, where the framework of the system is described in Section 3.1; onsite travel behavior data collecting method is explained in Section 3.2. The Tourist-Behavior pattern sequences mining method and tangible travel route recommendation procedure are thoroughly explained in Sections 3.3 and 3.4, respectively. Section 4 analyzes the experimental results to validate the feasibility and performance of the proposed system. The conclusion and future works are presented in Section 5.

2. Related Works

2.1. Travel Route Recommendation Systems. Due to the practical values of travel route recommendation systems, lots of researchers have been placing a great emphasis on solving the route planning in tourism scenarios [19, 23] in recent years. One category of these works uses the Orienteering problem [24] and its variants to approach the route planning problem. These methods formulate the problem from different perspectives, resulting in diverse problem models, which consider different problem variables and constraints. The route-generation process actually is a near-optimal solution using metaheuristic searching algorithm [25]. Accordingly, to enhance the personalization of the recommended routes, most of these works resort to acquiring more detailed user feedbacks or profiles to assist in fine-tuning the final results. In [26], the system solicits walking travel related attributes from tourists to insert concrete walking routes into POI itineraries, thereby supporting more experiential exploration of tourist destinations. Zhang et al. [27, 28] studied tour recommendation with the goal of recommending personalized itineraries based on the interest preferences of users and available touring time, while considering opening hours of POIs and uncertainty in travelling time. Other studies consider more practical factors that raise novel optimization challenges incorporating forms of situational awareness such as multiple modes of transport [29], considering traffic conditions [30–32], POI crowdedness [33, 34], and queuing times [35].

Although the optimization-based route planning systems can recommend a reasonable travel route adhering to one’s preferences and constraints, the interactive preferences input of the planning process is time costly to tourists. It is impractical for tourists to spend a long time in inputting a complex user profile when entering a specific POI. Furthermore, the results of these systems are lack of diversity and less personalization due to the near-optimal solution searching methodology. Therefore, lots of other works focus on generating personalized travel routes by mining User Generated Contents (UGC), that is, data-driven approaches to route planning. The UGC adopted in previous researches include GPS trajectory datasets [36], check-in datasets [37, 38], and geo-tagged photos [39].

Chen et al. [31] adopted historical check-in data and GPS trajectories to construct a POI network and used a heuristic method to generate a favorite POIs list for a specific user in an interactive manner. Subsequently, the system requests users to specify their favorite POIs during the route-generating stage. PersTour system [13, 40] uses geo-tagged photos to determine POI and construct POI travel sequences and leverage an Orienteering problem solving model to recommend POI itineraries by both considering POI popularity and tourist personal interests. Majid et al. [41] inferred the location of POIs and their semantic meaning using clustering approaches on geo-tagged photos and used a pattern mining
algorithm to discover popular travel sequences under the context of the tour recommendation, that is, time, day, and weather. Besides, the rapid growth of online tourism websites provides massive POI reviews and travelogues. Thus, some recent works [14, 42–44] focused on generating personalized mining travelogues and POI related contents.

The data-driven based route planning systems can recommend rather personalized and reasonable travel routes; however, the limitation of these systems is that they aim at recommending city-level or district-level orienting POIs itineraries, that is, planning POIs travel routes for tourists within a city or region. They fail to generate tangible travel routes for tourists within a specific POI due to lack of rich onsite travel behavior and related itinerary mining algorithms.

2.2. Personalized Recommendation in IoT Environment. The IoT concept was first coined by Kevin Ashton in 1999 [45] in supply chain management applications based on radio frequency identification devices (abbr. RFID). At present, IoT is referring to a bundle of technologies that aim at sensing, handling, and transmitting state information of physical environments, which is broadly applied in smart cities [46, 47], smart business [48, 49], and smart tourism scenarios [50]. The goal of these smart systems is to recommend a set of personalized and valuable items or services for various users. To this end, researchers focus on recording and analyzing user behavior to learn user preferences more precisely by deploying IoT technologies.

Specifically in smart tourism applications, some studies focus on using IoT technologies and mobile devices to improve tourism experiences in an interactive way. Kuusik et al. [51] designed a smart museum system that integrates PDAs and RFID technologies to provide users with cultural contents by sensing the interactive behavior between PDAs and RFID tags, which were installed near each artwork. In [52], an indoor location-aware system was designed for a smart museum to enhance visitors’ cultural experiences. The proposed system obtains visitors localization information through a Bluetooth low energy (BLE) infrastructure installed in the museum and uses several location-aware services hosted in the system to interact with visitors according to their locations.

And some other works aim at solving the next visiting spot recommendation problem within a specific POI. Massimo et al. [5] leveraged Inverse Reinforcement Learning method to learning user preferences by observing tourists onsite behavior in an IoT-equipped smart museum so as to predict next exhibit sequentially for tourists. Hashemi et al. [6, 7] solved the challenging next POI recommending problem by logging and mining users’ onsite physical and online interaction behavior data within an IoT-augmented museum. However, the above works fail to generate personalized and tangible travel routes for tourists.

To that end, some researchers strived to solve this challenging problem by mining historical travel trajectories in an IoT-augmented environment. Tsai et al. [15] adopted RFID infrastructure to record visitors’ check-in sequences of recreation facilities in a theme park and then proposed a statistical method to find behavioral similar historical visitors so as to suggest a travel route for the current querying visitor. Luo et al. [16] studied a new path finding system that discovers the most frequent path during user-specified time periods in large-scale historical trajectory data. Tsai et al. [17] proposed a touring path suggesting system for visitors to comprehend exhibits in exhibitions or museums. The system takes previous popular visiting trajectories as the suggestion foundation and provides a time-interval sequential patterns mining algorithm improved from [18] to generate personalized travel routes. However, as the above systems only resorted to dedicated IoT devices to record the check-in behavior, they hardly learn more tangible user preferences towards each interest object from the single dimensional behavior. Meanwhile, current smartphones generally equip a camera and diverse sensors, which could be used to sense multiple dimensional onsite behaviors of tourists so as to explore high-level tourist preferences and recommend personalized tangible travel route. Although some previous researches have investigated the human activity recognition based on smartphone sensors [21, 53], there is no study on learning user preferences directly by smartphones. To the best of our knowledge, our proposed system is the first work of leveraging smartphones and IoT environment to recommend tangible travel route within POIs based on onsite travel behavior sensing and mining methods.

3. Research Methodologies

3.1. System Overview. In this work, we use the phrase scenic area to denote a park or a museum containing a series of sightseeing spots or exhibits, namely, interesting spots. At each interesting spot, entrance and exit of a scenic area need to be preinstalled a Bluetooth low energy (BLE) beacon to locate tourists in an indoor or outdoor scenario. An illustrative example of the system is shown in Figure 1. Concretely, our system adopts iBeacon [22] devices to indicate specific spots by broadcasting their own device tags, that is, positioning information. When a tourist is approaching an interesting spot with a smart phone, the phone will use the locating information to judge whether the tourist has arrived at this spot. If so, the phone will record the onsite travel behavior data at this spot. At the end of the travel, the phone uploads a complete behavior sequence and a user-specified profile to the system server. Subsequently, the server preprocesses these data, transforming them into Tourist-Behavior (TB) pattern sequences, and uses the TB pattern mining algorithm to generate candidate tangible travel routes. At the recommendation stage, the system server will recommend personalized tangible travel routes for a new tourist by using the route ranking method according to the tourist’s personal profile and constraints. The recommended travel route, which contains a spot visit sequence and their respective visit durations, will help them to finish a valuable tour in the area in a comfortable way.

Figure 2 illustrates the workflow of the proposed system. Specifically, stage 1 is performed by a client App running on tourists’ smart phones, which is responsible for collecting
tourists’ personal profiles and their behavior data, while stages 2 and 3 are performed on the system server side. In offline running stage 2, behavior sequence preprocessing method and Tourist-Behavior (TB) PrefixSpan algorithm are proposed to generate a series of TB pattern sequences, that is, candidate tangible travel routes. In online running stage 3, the system server recommends tangible travel routes for various tourists based on their profiles and route constraints.

3.2. Onsite Travel Behavior Collecting. Since tourists with different personal attributes may have different personal interests, stamina, walking speeds, and so forth to ensure the personalization of our recommendations, we classify and store the collected behavior sequences according to corresponding personal profiles in our system. At the beginning of the behavior data collection process, we request each tourist to input three common and typical personal attributes, including gender, age group, and education level, as a simple profile. Then, the client App uploads an onsite travel behavior sequence and a corresponding profile together to the system server. Subsequently, at the recommendation stage, the system server uses a personal profile of the querying tourist to retrieve generated routes from the corresponding route subset for matching tourists’ different interests.

3.2.1. Positioning Mechanism. The positioning mechanism is implemented based on iBeacon devices and smartphones. The iBeacon protocol is characterized as low energy
consumption and broad wireless broadcasting range, which can be applied in indoor and outdoor scenarios. Besides, there is no pairing connection during the locating process, which differs from traditional Bluetooth protocols. Therefore, iBeacon makes the positioning mechanism more flexible and efficient.

During the tourist locating process, the iBeacon devices constantly broadcast their own location identities (ID) with a TX power value. The positioning information consists of two 16-bit protocol data fields, named major ID and minor ID, which are used to represent a scenic area and an interesting spot, respectively. Meanwhile, a nearby smartphone adopts (1) to compute a proximity distance \( d \) between itself and the broadcasting iBeacon device to locate itself in a scenic area.

\[
    d = 10^\left(\frac{\text{RSSI} - A}{10 \times n}\right),
\]

where \( A \) is the TX power constant that stands for the received signal strength at 1-meter distance from the iBeacon device; RSSI is the current BLE signal strength of the smart phone; \( n \) is the path loss coefficient constant; and \( d \) is a distance of meters between the smartphone and the iBeacon device [54]. The client App running on a smartphone chooses the lowest \( d \) as the current recognized interesting spot when the phone is receiving multiple iBeacon signals simultaneously.

### 3.2.2. Travel Behavior Sensing and Recording

During the onsite behavior sensing procedure, the client App has two tasks: (a) reckoning the current interesting spot where the tourist is arriving at, meanwhile recording the arriving and leaving timestamps of each interesting spot by comparing the distance threshold with the real distance between the current iBeacon device and the smartphone. (b) Monitoring the data of smartphone devices, that is, the on-board camera and accelerometer, so as to record the behavior of taking pictures and standing still to appreciate something on each interesting spot of the tourist.

To record the number of taking pictures behaviors, the client App monitors the on-board camera operation message of Android system, namely, "android.hardware.action.NEW_PICTURE", once the tourist uses the phone camera to take a picture. To record the number of standing behaviors, the client App integrates 3-dimensional accelerations into an overall acceleration data first. Then, it uses a Sliding Window Filtering method [55] to count the number of standing behaviors. The client App inserts the number of these two behaviors into the current travel behavior sequence. Last, the client App uploads the behavior sequence and its corresponding profile to the system server when it detects the exit of the scenic area.

Let \( B = \{b_1, b_2, \ldots, b_g\} \) be the set of iBeacon devices that are installed in a specific scenic area. In the system server, a travel behavior sequence record is stored as \( <\text{s}i\text{d}, \text{t}bs> \), where \( \text{s}i\text{d} \) is the identifier of the sequence and \( \text{t}bs \) is an onsite behavior sequence. And \( \text{t}bs \) consists of a sequence \( <\text{st}i\text{ni}, b_i, \text{st}o\text{i}, p_i, s_i, s_t> <\text{st}i\text{ni}, b_i, \text{st}o\text{i}, p_i, s_i> \ldots <\text{st}i\text{ni}, b_i, \text{st}o\text{i}, p_i, s_i> \), where the quintuple \( <\text{st}i\text{ni}, b_i, \text{st}o\text{i}, p_i, s_i> \) represents a behavior data with respect to the interesting spot \( i; b_i \) is the corresponding iBeacon device ID and \( b_i \in B; \text{st}i\text{ni} \) and \( \text{st}o\text{i} \) stands for arriving and leaving timestamps, respectively, and \( \text{st}i\text{ni} \leq \text{st}o\text{i}, \text{st}n_{i+1} \) for \( 1 \leq i \leq k-1; p_i \) and \( s_i \) are the number of taking pictures and standing still to appreciate something, respectively. Further, the visit duration of spot \( i \) is calculated by \( \text{st}o\text{i} - \text{st}i\text{ni} \); the interval between spot \( i \) and spot \( i+1 \) is calculated by \( \text{st}i\text{ni}_{i+1} - \text{st}o\text{i}_{i} \).

Example 1. As illustrated in Figure 1, there are one entrance, one exit, and seven interesting spots in the scenic area. Thus, there are nine iBeacon devices as total needed to install in the area. After tourist #4 inputs his or her profile and time constraint, the system returns a travel route by mining the historical travel behavior sequences acquired from the other three tourists. The corresponding sequences are shown in Table 1; for example, tourist #1 visited six interesting spots \( A, B, D, F, E, \) and \( G \). The symbols \( Z_t \) and \( Z_e \) stand for the entrance and the exit, respectively. Taking the behavior data at spot \( A \) as an instance, tourist #1 arrived at spot \( A \) at the 6th min and left out at the 26th min, took 5 pictures, and stood still for 4 times at spot \( A \).

### 3.3. Tourist-Behavior Mining

The goal of the Tourist-Behavior mining stage is to generate various candidate travel routes by mining the historical onsite travel behavior sequences. This stage consists of two steps: the travel behavior sequence preprocessing step and the Tourist-Behavior sequential travel routes generating step.

#### 3.3.1. Travel Behavior Sequence Preprocessing

The preprocessing step is to transform travel behavior sequences into Tourist-Behavior (TB) pattern sequences and then store pattern sequences into route subset according to their corresponding personal profile. Before describing the details of the step, the following definitions are given.

**Definition 2.** A Tourist-Behavior (TB) pattern \( \lambda_i \) is defined as a triple \( <b_i, \text{NP}_i, D_i> \), where \( b_i \) is the location identity of spot \( i; \text{NP}_i \) is the normalized popularity value about spot \( i; D_i \) is the discrete visit duration at spot \( i \). Note that the pattern \( \lambda_i \) is said to match the pattern \( \lambda_j \) if and only if \( b_i = b_j, \text{NP}_i = \text{NP}_j, \) and \( D_i = D_j \).
Definition 3. Let \( \Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_x\} \) be the set of TB patterns and let \( \delta_i \) be the discrete interspot travel time in a travel behavior sequence. A sequence \( \alpha = (V_1, \delta_1, V_2, \delta_2, \ldots, \delta_{k-1}, V_k) \) is a TB sequence if \( V_s \in \Lambda \) for \( 1 \leq s \leq h \) and \( \delta_i = \text{Disc}(\Delta t) \) for \( 1 \leq s \leq h - 1 \).

First, the preprocessing method cleans up the passing-by behavior data and calculates the interspot travel time and the visit duration in each travel behavior sequence. Hence, the method needs to delete the behavior data if the visit duration is shorter than a time threshold \( T_v \), except for the entrance and exit behavior data. Let \( \delta_i \) be the discrete interspot travel time for the tourist to travel from spot \( i \) to spot \( i+1 \), and let \( D_i \) be the visit duration at spot \( i \); \( \Delta t \) stands for \( \delta_i \) or \( D_i \); and \( T_d \) is the metric of the discrete time. Consequently, the discrete time integer of \( \delta_i \) and \( D_i \) can be derived from the following equation:

\[
\text{Disc}(\Delta t) = \left\lceil \frac{\Delta t}{T_d} \right\rceil \quad (2)
\]

Second, the method calculates popularity values of each interesting spot in each travel behavior sequence. As each travel behavior sequence is collected from an individual tourist, two popularity values of the same spot in two sequences are probably different due to two different tourists’ onsite behaviors. The prior knowledge of the method is that tourists will spend longer visit duration, take more pictures, or stand still more times to appreciate something at a spot if they are more interested in the spot. The popularity value of spot \( i \) in a specific sequence can be calculated by the following equation:

\[
\text{Pop}_i = \frac{w_1 \times \text{stout}_i - \text{stin}_i}{\sum_{j=1}^{k} (\text{stout}_j - \text{stin}_j)} + \frac{w_2 \times \text{pi}_i}{\sum_{j=1}^{k} \text{pi}_j} + w_3 \\
\times \frac{s_i}{\sum_{j=1}^{k} s_j} \quad (3)
\]

where \( w_1, w_2, \) and \( w_3 \) are weights used to calculate \( \text{Pop}_i \) and \( w_1 + w_2 + w_3 = 1 \); for each travel behavior sequence, the total visit duration is derived from \( \sum_{i=1}^{k} (\text{stout}_i - \text{stin}_i) \); \( \sum_{i=1}^{k} \text{pi}_i \) and \( \sum_{i=1}^{k} s_i \) denote the total number of times of taking pictures and standing still within the sequence, respectively. Besides, to make the popularity values of spots in different sequences comparable, we normalize all popularity values in each sequence. In detail, to calculate a normalized popularity value \( \text{NP}_i \) of spot \( i \), all spots in each sequence are ranked as a descending list according to their respective \( \text{Pop}_i \). To calculate \( \text{NP}_i \), the list is divided into \( n \) segments where \( n \) denotes the popularity normalization coefficient. For example, in (4), the normalization coefficient is to be 4; all spots ranking in the top \( l/n \) in a specific sequence are assigned with a normalized popularity value of \( L_n \), indicating that the querying tourist is most likely to be interested in these spots.

\[
\text{NP}_i = \begin{cases} 
L_n, & \text{if Pop ranks in the top } \frac{l}{n} \\
& \vdots \\
L_2, & \text{if Pop ranks in } \left[ \frac{n-2}{n} \frac{n-1}{n} \right) \\
L_1, & \text{if Pop ranks in the lowest } \frac{1}{n} 
\end{cases} \quad (4)
\]

After the preprocessing step, all travel behavior sequences are transformed into TB pattern sequences and stored in the Tourist-Behavior sequence database (abbr. TBD), according to their respective profiles. Specifically, our system divides tourist age into three groups, below 20 years, from 20 to 55 years, and over 55 years, and classifies education into three levels: preundergraduate, undergraduate, and graduate. By multiplying with two gender attributes, there are \( 3 \times 3 \times 2 = 18 \) TBDs in total in our system with the above three profile attributes.

Example 4. Let us take the travel behavior sequences shown in Table 1 as an example to explain the travel behavior sequence preprocessing method. Suppose that \( T_v \) is set at 5 minutes; \( T_d \) is set at 10 minutes; and \( w_1, w_2, \) and \( w_3 \) are set as 0.4, 0.3, and 0.3, respectively. At first, the behavior data \( <99, E, 110, 0, 0> \) is deleted as a passing-by behavior data in \( \text{sid 01} \), because its visit duration is shorter than \( T_v \). Further, the interspot travel time and the visit duration are discretized by (2). Next, the popularity of each spot is computed; for example, \( \text{Pop}_{A1} \) with respect to tourist \#1 is calculated as \( 0.4 \times 20/82 + 0.3 \times 5/14 + 0.3 \times 4/18 = 0.271 \); the visit duration \( D_A \) is 20 minutes; the total visit duration is 82 minutes; the number of taking pictures and standing still is \( I_4 \) and \( I_8 \), respectively. The corresponding TB pattern sequences are shown in Table 2.

3.3.2. Tourist-Behavior Sequential Travel Routes Generating.
As the onsite travel behaviors are complex and contain noisy behavior data, for example, one making a phone call or taking a sit for a break during a visit, we need a method to discover popular travel routes and to filter noisy travel behaviors. Therefore, we design the TB PrefixSpan algorithm to discover all frequent TB patterns with the corresponding interspot travel time and to construct various Tourist-Behavior (TB) sequential travel routes from a TBD. An improvement of the TB PrefixSpan algorithm compared to [54] is that due to the fact that TB pattern sequences separately contain discrete interspot travel time and spot visit durations, the TB PrefixSpan algorithm can delete visit durations of non-frequent TB patterns yet preserve intervals to ensure the accurate time arrangement of new TB sequential patterns. Before describing the TB PrefixSpan algorithm, the following definitions are given.

Definition 5. Assume two TB pattern sequences \( \alpha = (V_{a1}, \delta_{a1}, V_{a2}, \delta_{a2}, \ldots, \delta_{(k-1)}, V_{ak}) \) and \( \beta = (V_{b1}, \delta_{b1}, V_{b2}, \delta_{b2}, \ldots, \delta_{(h-1)}, V_{bk}) \) \( (h \leq k) \); \( \beta \) is said to be contained in \( \alpha \) or a TB subsequence of \( \alpha \); that is, \( \beta \subseteq \alpha \), if there exist sequence
Table 2: An example of Tourist-Behavior pattern sequence.

<table>
<thead>
<tr>
<th>Sid</th>
<th>Tourist-Behavior sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L4, 3&gt;, \langle C, L3, 2&gt;, \langle E, L3, 2&gt;, \langle I, G, L4, 2&gt;, \langle H, L4, 3&gt;, \langle I, L2, 2&gt;, \langle I, X, L2, 3&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
<tr>
<td>02</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L4, 3&gt;, \langle B, L2, 2&gt;, \langle C, L1, 2&gt;, \langle F, L3, 2&gt;, \langle G, L3, 2&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
<tr>
<td>03</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L2, 2&gt;, \langle C, L3, 2&gt;, \langle E, L3, 2&gt;, \langle I, G, L4, 2&gt;, \langle I, X, L3, 2&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
</tbody>
</table>

Table 3: An example of TB sequences database.

<table>
<thead>
<tr>
<th>Sid</th>
<th>TB pattern sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L4, 3&gt;, \langle C, L3, 2&gt;, \langle E, L3, 2&gt;, \langle I, G, L4, 2&gt;, \langle H, L4, 3&gt;, \langle I, L2, 2&gt;, \langle I, X, L2, 3&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
<tr>
<td>02</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L4, 3&gt;, \langle B, L2, 2&gt;, \langle C, L1, 2&gt;, \langle F, L3, 2&gt;, \langle G, L3, 2&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
<tr>
<td>03</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L2, 2&gt;, \langle C, L3, 2&gt;, \langle E, L3, 2&gt;, \langle I, G, L4, 2&gt;, \langle I, X, L3, 2&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
<tr>
<td>04</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L4, 3&gt;, \langle C, L3, 2&gt;, \langle E, L3, 2&gt;, \langle I, G, L4, 2&gt;, \langle I, X, L3, 2&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
<tr>
<td>05</td>
<td>(&lt;Ze, L1, I&gt;, \langle A, L4, 3&gt;, \langle C, L3, 2&gt;, \langle E, L3, 2&gt;, \langle I, G, L4, 2&gt;, \langle I, X, L3, 2&gt;, \langle I, Zt, L1, I&gt;)</td>
</tr>
</tbody>
</table>

indices \(1 \leq j_1 < j_2 < \cdots < j_h \leq k\) such that (1) \(V_{\beta_1} = V_{a_1}\) and \(V_{\beta_2} = V_{a_2}, \ldots, V_{\beta_h} = V_{a_h}\); (2) \(\delta_{\beta_1} = \delta_{a_1}\) and \(\delta_{\beta_2} = \delta_{a_2}, \ldots, \delta_{\beta_h} = \delta_{a_h}\).

Definition 6. A TB pattern \(\gamma\) is called a frequent TB pattern if the number of sequences in a TBD, which contains \(\gamma\) as the subsequence, is greater than or equal to the user-specified minimum support, called \(\minsup\) or \(\minsup\). That is, \(\gamma\) is called a frequent TB pattern in a TBD if \(\sup\) \(\gamma \geq \minsup\). The TB postfix of \(\gamma\) is named a projection of \(\gamma\) with respect to \(\alpha\), is \(TB\) prefix of \(\gamma\) in a TBD if \(\sup\) \(\gamma \geq \minsup\), and \(\sup\) \(\gamma \geq \minsup\), where \(\minsup\) is an accumulated time from spot \(i\) to spot \(k\); that is, \(\delta_N = \delta_i + \delta_i + \ldots + \delta_{k-1}\).

Specifically, the algorithm initially recognizes each frequent TB pattern to construct their corresponding \(\alpha\)-projection databases. For each TBD\(_{\alpha}\) database, the algorithm constructs the corresponding TBTable to identify all frequent table cells. Then, for each frequent cell the element (\(\delta_N, \lambda_j\)) is appended to the end of \(\alpha\) to construct a new TB prefix \(\alpha'\) and then the \(\alpha'\)-projection database TBD\(_{\alpha'}\) is built. Recursively constructing all of the frequent TB pattern sequences in the TBD\(_{\alpha'}\) discovers all TB sequential travel routes in the TBD, which are stored in the TB sequential travel route database (abbr. TBSTR).

Example II. Let us take five TB pattern sequences in Table 3 as an example to explain the TB sequential travel route mining process, where the \(\minsup\) is set at 2. The TB PrefixSpan algorithm can also be deemed as a Tree Traversal algorithm; each node of the growing trees corresponds to a TBTable\(_{\alpha}\). As shown in Figure 4, we use red solid arrows to illustrate steps of constructing a TB sequential travel route \(<Ze, L1, I>, I, A, L4, 3>, 2, F, L2, 3>, 4, X, L3, 2, I, Zt, L1, I>\. In the beginning, the algorithm constructs TBTable\(_{\alpha'}\), which recognizes 16 1-length sequential routes listed in Table 4. Suppose that the algorithm is mining the current 3-length route \(<Ze, L1, I>, I, A, L4, 3>, 2, F, L2, 3>\. Thus TBTable\(_{F,L2,3}\) needs to be constructed subsequently. As shown in Table 5, the \(<F, L2, 3>\) projection database has two TB postfix \(\alpha'\) and \(\alpha''\). The results of TBTable\(_{F,L2,3}\) are shown in Table 6. There are 3 frequent TB patterns labeled in bold in the table, that is, \(H, L2, 2\) and \(X, L3, 2\), \(<Ze, L1, I>, I, A, L4, 3>, 2, F, L2, 3>\).
Tourist-Behavior sequential travel route mining algorithm
Subroutine TB-PrefixSpan($\alpha$, $l$, $TBD|_e$)
Predefine symbols:

$\text{min\_sup\_count}$  Minimum support count
$A$  A set of frequent TB patterns
$\text{TBSTR}$  The TB sequential travel route database

Input:
$\alpha$  A $l$-length TB sequential travel route
$l$  The length of $\alpha$
$TBD|_e$  The $\alpha$–projection database in TBD

Output:
$\alpha'$  A $(l+1)$-length TB sequential travel route

Method:
(1)  if $l = 0$ then
    (2)  read the set of TB patterns $A$;
    (3)  for each frequent TB pattern $\lambda_i$ in $A$
      (4)  append $\lambda_i$ to $\alpha$ as $\alpha'$;
      (5)  store $\alpha'$ into $\text{TBSTR}$;
      (6)  construct $\alpha'$–projection database $TBD|_\alpha$;
      (7)  call subroutine TB-PrefixSpan($\alpha'$, $l+1$, $TBD|_e$);
    (8)  end for
(9)  else
(10)  build an empty TB_table;
(11)  for each sid in $TBD|_e$
      (12)  set new interval variable $\delta_N$ to 0;
      (13)  for each element $(\delta_i, \lambda_i)$ of the sid in $TBD|_e$
        (14)  accumulate interval, i.e., $\delta_N = \delta_N + \delta_i$;
      (15)  increase the count of cell $(\delta_N, \lambda_i)$ by 1;
    (16)  end for
(17)  end for
(18)  for each cell $(\delta_i, \lambda_i)$ in the TB_Table
      (19)  if the count of cell $(\delta_i, \lambda_i)$ $\geq \text{min\_sup\_count}$ then
          (20)  append $(\delta_i, \lambda_i)$ to $\alpha$ as $\alpha'$;
          (21)  store $\alpha'$ into $\text{TBSTR}$;
          (22)  construct $\alpha'$–projection database $TBD|_\alpha$;
          (23)  call subroutine TB-PrefixSpan($\alpha'$, $l+1$, $TBD|_e$);
      (24)  end if
    (25)  end for
(26)  end if

Figure 3: The pseudocode of the TB PrefixSpan algorithm.

and $\langle Zt, L1, L1 \rangle$. The algorithm joins these 3 patterns behind the current route, respectively, to construct 3 new 4-length routes, and then constructs 3 corresponding TB_tables of these patterns. The algorithm recursively traverses the tree to discover all potential TB sequential travel routes.

3.4. Travel Route Ranking and Recommending. As mentioned in Section 3.3.1, all TB pattern sequences are divided into subsets according to the personal profile after the sequence preprocess step. To make the recommended routes match the querying tourist’s personal interests and characteristics better, we design a route ranking method to search valuable

<table>
<thead>
<tr>
<th>TB pattern</th>
<th>Sup_count</th>
<th>TB pattern</th>
<th>Sup_count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle Ze, L1, L1 \rangle$</td>
<td>5</td>
<td>$\langle H, L2, 2_1 \rangle$</td>
<td>2</td>
</tr>
<tr>
<td>$\langle A, L4, 3 \rangle$</td>
<td>3</td>
<td>$\langle H, L2, 3 \rangle$</td>
<td>2</td>
</tr>
<tr>
<td>$\langle B, L3, 3 \rangle$</td>
<td>2</td>
<td>$\langle L, L1, 1 \rangle$</td>
<td>2</td>
</tr>
<tr>
<td>$\langle C, L3, 1 \rangle$</td>
<td>2</td>
<td>$\langle L, L4, 2 \rangle$</td>
<td>2</td>
</tr>
<tr>
<td>$\langle D, L3, 2 \rangle$</td>
<td>2</td>
<td>$\langle X, L2, 3 \rangle$</td>
<td>3</td>
</tr>
<tr>
<td>$\langle E, L3, 2 \rangle$</td>
<td>2</td>
<td>$\langle X, L3, 2 \rangle$</td>
<td>2</td>
</tr>
<tr>
<td>$\langle F, L2, 3 \rangle$</td>
<td>2</td>
<td>$\langle R, L3, 3 \rangle$</td>
<td>2</td>
</tr>
<tr>
<td>$\langle G, L4, 2 \rangle$</td>
<td>2</td>
<td>$\langle Zt, L1, L1 \rangle$</td>
<td>5</td>
</tr>
</tbody>
</table>
and reasonable routes from a TBSTR matched by an input personal profile.

Thus, the method first requests the querying tourist to input a personal profile and a route constraint. The route constraint includes the intended travel duration and specified travel start and end location of the POI. Next, the method uses the personal profile to retrieve candidate TB sequential travel routes in the corresponding TBSTR. After retrieving candidate TB sequential travel routes, the server filters out travel routes that do not meet the input route constraint. In detail, the server reserves the routes of which start and end points match the user-specified entrance and exit. This is for considering the situation of multiple entrances and exits existing in one scenic area. Then, it adds up the total route duration time $T_{ta}$ of the route $\alpha$ by using the following equation:

$$T_{ta} = \left( \sum_{i=1}^{\lvert \alpha \rvert - 1} \delta_i + \sum_{i=1}^{\lvert \alpha \rvert} D_i \right) \times T_{d},$$  \hspace{1cm} (5)$$

where $\lvert \alpha \rvert$ is the length of the route $\alpha$; $\delta_i$ and $D_i$ are the $i$th discrete interval and spot visit duration of the route $\alpha$, respectively; $T_d$ is the metric of the discrete time. And the server selects the candidate routes that meet the following equation:

$$(1 - \varphi) \times ITD \leq T_{ta} \leq ITD,$$  \hspace{1cm} (6)$$where $ITD$ stands for an intended travel duration of the querying tourist; $\varphi$ is a filtering condition parameter between $[0, 1]$ used to set a filtering range for the candidate routes.

At last, the system server recommends the most valuable Top-k tangible travel routes for the querying tourist by calculating route values of the remaining routes. A route value consists of the total normalized popularity value and the ratio of the total visit duration to the total route duration. The candidate travel routes set is denoted as $\{CR_i \mid i = 1, 2, ..., M\},$
where $M$ is the total number of candidates. The route value $RV_\alpha$ of route $\alpha$ is calculated by the following equation:

$$RV_\alpha = w_{pl} \times MMN \left( \sum_{i=1}^{p_i} np_i \right) + w_{rd} \times \left( \frac{\sum_{i=1}^{p_i} D_i}{\sum_{i=1}^{p_i-1} \delta_i + \sum_{i=1}^{p_i} D_i} \right)$$

(7)

where $w_{pl}$ and $w_{rd}$ are the weights used to calculate $RV_\alpha$ and $w_{pl} + w_{rd} = 1$. $\sum_{i=1}^{p_i} np_i$ is the total normalized popularity value of route $\alpha$; $\left( \frac{\sum_{i=1}^{p_i} D_i}{\sum_{i=1}^{p_i-1} \delta_i + \sum_{i=1}^{p_i} D_i} \right)$ is the ratio of the total visit duration to the total route duration of route $\alpha$; $MMN(\cdot)$ is min-max normalization function for normalizing the rank value among $\{CR_i | i = 1, 2, \ldots, M\}$.

4. Experiment and Discussion

In this section, we designed a validation experiment and several performance analysis experiments to test our system. The iBeacon devices were based on CC2541 embedded processor developed by Texas Instruments Company. The client App was developed with Android Studio 2.3.3, which runs on Android smartphone system version 6.0.1 upwards. The system server runs on a workstation with Intel Xeon 3.5GHz and 16 GB RAM, and related application was developed by python version 2.7.13 running on Ubuntu 14.04 with Tomcat 6.0.

### 4.1 Validation Experiment

The primary goals of our empirical experiment are to (1) examine how well the recommended routes match a tourist’s actual interests, (2) demonstrate the route value of the top recommended routes, and (3) analyze the rationality of the top recommended routes. In this section, we introduce the experimental settings of the validation experiment and present the results of a test of onsite behavior data collection. Last, we present the results of the validation experiment to validate the ability of our system in recommending personalized tangible travel routes for tourists in a given POI based on historical onsite travel behavior.

At first, we deployed our system in a small experimental exhibition hall with 20 exhibits, where each exhibit is preinstalled with an iBeacon device. And we invited 20 male and 20 female undergraduate students as volunteers to visit the experimental hall so as to collect their onsite travel behavior data. The layout of the experimental hall is shown as in Figure 6, in which topical-related posters are exhibited with a similar length. In detail, from B1 to B5 are scientific topical posters, from B6 to B12 are sports-related, and from B13 to B20 are daily life related contents. And we set the minimum support count of TB PrefixSpan at 2 and set discrete time metric $T_d$ at 1 minute; the popularity normalization coefficient is to be 5; the filtering condition parameter $\varphi$ is set at 0.2; the weights $w_{pl}$ and $w_{rd}$ in (7) were equally set at 0.5.

Figure 5 illustrates a test of onsite behavior acquisition. Figure 5(a) shows the experimental environment in which one volunteer carrying a smartphone is visiting an exhibit that is labeled by an iBeacon device. Figure 5(b) shows the software interface of the client App, which is selecting the nearest iBeacon device as the recognized spot, that is, Minor device, to collect following onsite behavior data. Figure 5(c) presents the original behavior sequence gathered on the Minor spot.

To examine how well the recommended routes match a tourist’s actual interests, we requested the volunteers to fill a rating questionnaire regarding the exhibiting posters after visiting the exhibition hall so as to directly learn interests of female and male volunteers. Table 9 lists the most favorite 10 posters for female and male volunteers, respectively, which reflects that female volunteers prefer daily life topical
posters, while male volunteers prefer sports-related contents. Next, we assumed two querying tourists’ route constraints and personal profiles that are listed in Table 7 to request route recommendations from our system. The top 3 valuable tangible travel routes recommended to two tourists are listed in Table 8. It can be easily observed from Table 8 that all candidate routes comply with corresponding tourist’s time constraints. More importantly, by comparing the posters of Tables 8 and 9, we find that about 80% of top 10 favorite posters regarding the two corresponding tourists are included in both top recommended routes. For instance, the first route recommended to the female tourist suggests she spends a relatively longer time at B2, B15, and B16 posters which are the top favorite posters to female listed in Table 9, while the first route recommended to the male tourist recommends B4, B6, and B7 posters. This observation proves that our system can learn different personal preferences from real onsite travel behaviors.

To demonstrate the route value of the top recommended routes, our route ranking method ranks top 3 valuable tangible travel routes which are listed in Table 8. It can be easily observed in Table 8 that both top 1 routes recommended to young female and male tourists have the biggest route value. For example, compared to the rest two routes, the top 1 route recommended to young male tourist possesses the largest total normalized popularity value (i.e., \(L/8\)), the longest total visit duration (i.e., 35 minutes), and the largest number of visit spots (i.e., 8 spots).

Regarding the rationality of the top recommended routes, as the recommended tangible routes are generated from onsite travel behaviors of young female or male tourists, the visit arrangements of the recommended route (e.g., the visit sequence, the interspot travel time, and the spot visit duration) can completely comply with the layout of the exhibition hall. As a result, the recommended routes possess rather visit rationality. For instance, the interspot travel time of a tangible route recommended to young females is derived from the average walking speed of young females; and the visit duration of each spot is calculated from the historical onsite travel behaviors of young females, for example, the average reading speed of young females. Figure 6 illustrates the visit sequence of two top 1 tangible routes for two querying tourists. The results indicate that the recommended route can help two querying tourists to finish their respective time-limited visits comfortably.

### Table 7: Two querying tourists’ route constraints and personal profiles

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Time constraint</th>
<th>Personal profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>45 min</td>
<td>Young/Female/Undergraduate</td>
</tr>
<tr>
<td>#2</td>
<td>30 min</td>
<td>Young/Male/Undergraduate</td>
</tr>
</tbody>
</table>

#### 4.2. Algorithm Performance Analysis

In this section, we study the performance of our TB PrefixSpan algorithm under different parameters settings. Obviously, longer travel routes, which contain more interesting spots, can meet longer travel duration query needs. Meanwhile, larger number of generated travel routes can provide more diverse personal recommendations for tourists. Thus, the length and the quantity of generated TB sequential travel routes reflect the effectiveness and quality of the recommendations in this work. Furthermore, to test the scalability of our algorithm, we construct a synthetic data set consisting of 12,000 randomly generated travel behavior sequences. All of the experimental travel behavior sequences are randomly selected from the synthetic data set. Without any other notice, the following experimental parameters settings are the same as the validation experiment.

#### 4.2.1. Data Size of Travel Behavior Sequences

To comprehend how the number of travel behavior sequences (data size) affects the TB sequential travel route generation, the data size is changed from 500 to 2,500 and the \(minsup\) is set at 4. Figure 7 demonstrates the average length and the longest length of the generated routes under different data sizes. As the data size is getting bigger, the length of TB sequential travel routes is getting longer; that is, the quality of routes is getting better. Therefore, the more historical onsite
Table 8: Top 3 candidate tangible travel routes generated for two tourists.

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Candidate tangible travel routes</th>
<th>Total route duration</th>
<th>Total visit duration</th>
<th>Total exhibits</th>
<th>Total NP,</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>&lt;E0, L1, , 1, &lt;B1, L4, 4, 1, &lt;B2, L5, 5, 0, &lt;B3, L4, 3, 1, &lt;B6, L4, 2, 2, &lt;B14, L5, 4, 1, &lt;B15, L5, 6, 0, &lt;B16, L5, 6, 1, &lt;B17, L4, 5, 1, &lt;E1, L1, &gt;</td>
<td>43 min</td>
<td>35 min</td>
<td>8</td>
<td>1.38</td>
</tr>
<tr>
<td>#1</td>
<td>&lt;E0, L1, , 1, &lt;B2, L5, 5, 1, &lt;B3, L4, 3, 1, &lt;B12, L4, 2, 2, &lt;B14, L5, 4, 1, &lt;B15, L5, 6, 1, &lt;B17, L4, 5, 1, &lt;B16, L5, 6, 1, &lt;E1, L1, &gt;</td>
<td>40 min</td>
<td>31 min</td>
<td>7</td>
<td>1.34</td>
</tr>
<tr>
<td>#1</td>
<td>&lt;E0, L1, , 1, &lt;B1, L4, 4, , 1, &lt;B4, L4, 4, 0, &lt;B5, L3, 2, 3, &lt;B14, L5, 4, 1, &lt;B15, L5, 6, 0, &lt;B16, L5, 6, 1, &lt;B20, L3, 2, 1, &lt;E1, L1, &gt;</td>
<td>36 min</td>
<td>28 min</td>
<td>7</td>
<td>1.31</td>
</tr>
<tr>
<td>#2</td>
<td>&lt;E0, L1, , 1, &lt;B4, L5, 4, 0, &lt;B5, L4, 3, 1, &lt;B6, L5, 4, 0, &lt;B7, L5, 5, 0, &lt;B12, L5, 4, 0, &lt;B14, L4, 3, 0, &lt;B7, L4, 2, 1, &lt;E1, L1, &gt;</td>
<td>28 min</td>
<td>25 min</td>
<td>7</td>
<td>1.34</td>
</tr>
<tr>
<td>#2</td>
<td>&lt;E0, L1, , 1, &lt;B4, L5, 4, 0, &lt;B6, L5, 4, 0, &lt;B7, L5, 5, 0, &lt;B12, L5, 4, 0, &lt;B4, L4, 3, 1, &lt;B15, L4, 2, 2, &lt;E1, L1, &gt;</td>
<td>26 min</td>
<td>22 min</td>
<td>6</td>
<td>1.30</td>
</tr>
<tr>
<td>#2</td>
<td>&lt;E0, L1, , 1, &lt;B4, L5, 4, 0, &lt;B6, L5, 4, 0, &lt;B9, L5, 6, 1, &lt;B8, L4, 3, 1, &lt;B17, L4, 2, 1, &lt;E1, L1, &gt;</td>
<td>23 min</td>
<td>19 min</td>
<td>5</td>
<td>1.25</td>
</tr>
</tbody>
</table>
Table 9: The top 10 favorite posters of two types of volunteers.

<table>
<thead>
<tr>
<th>Poster No.</th>
<th>Poster Topic</th>
<th>Poster No.</th>
<th>Poster Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2</td>
<td>Scientific</td>
<td>B4</td>
<td>Scientific</td>
</tr>
<tr>
<td>B14</td>
<td>Daily life</td>
<td>B6</td>
<td>Sport</td>
</tr>
<tr>
<td>B1</td>
<td>Scientific</td>
<td>B7</td>
<td>Sport</td>
</tr>
<tr>
<td>B3</td>
<td>Scientific</td>
<td>B12</td>
<td>Sport</td>
</tr>
<tr>
<td>B15</td>
<td>Daily life</td>
<td>B9</td>
<td>Sport</td>
</tr>
<tr>
<td>B16</td>
<td>Daily life</td>
<td>B5</td>
<td>Scientific</td>
</tr>
<tr>
<td>B17</td>
<td>Daily life</td>
<td>B8</td>
<td>Sport</td>
</tr>
<tr>
<td>B19</td>
<td>Daily life</td>
<td>B14</td>
<td>Daily life</td>
</tr>
<tr>
<td>B12</td>
<td>Sport</td>
<td>B2</td>
<td>Scientific</td>
</tr>
<tr>
<td>B11</td>
<td>Sport</td>
<td>B17</td>
<td>Daily life</td>
</tr>
</tbody>
</table>

travel behavior the system gets, the higher quality of the recommendation can be generated.

4.2.2. Minimum Support of TB PrefixSpan. To discover how the minimum support parameter \( \text{min}_\text{sup} \) of TB PrefixSpan affects the quality of the TB sequential travel route, the different \( \text{min}_\text{sup} \) varying from 0.02% to 0.1% are tested with the synthetic data set.

Figure 8 presents the average length and the longest length of the generated routes under different \( \text{min}_\text{sup} \). As the minimum support increases, the length of generated routes decreases. If the \( \text{min}_\text{sup} \) is set at 0.02%, the average length of routes is 5.14 and the longest route contains 11 visit spots. However, if the \( \text{min}_\text{sup} \) is set at 0.1%, the average length of routes declines to 3.98, and the longest route only contains 6 visit spots.

Figure 9 shows the execution time of the TB PrefixSpan algorithm under different minimum supports. As the \( \text{min}_\text{sup} \) increases from 0.02% to 0.1%, the execution time of the TB sequential travel route generation process declines from 950.59s to 200.5s. Based on the observation from Figures 8 and 9, the \( \text{min}_\text{sup} \) is suggested as 0.04% or below to ensure the quality of the routes. As the algorithm is performed in an offline stage on the system server side, the running time of the algorithm will not affect the reaction speed of the online recommendation process.

4.2.3. Information Granularity of Tourist-Behavior Pattern. According to Definition 2, each TB pattern includes a location identity, a normalized popularity value, and visit duration. Thus, the information granularity of a TB pattern, which is affected by the metric of the discrete time and the popularity normalization coefficient, consequently decides the quality of the generated travel routes. To observe the relationship between the information granularity and the route quality, that is, the average length and longest length of the generated TB sequential travel routes, the following experiments are conducted.

Regarding the metric of the discrete time \( T_d \), a series of values ranging from 5 min to 20 min are tested by 2,000 sequences randomly selected from the data set. With different \( T_d \) settings, the average length and the longest length of the generated routes are shown in Figure 10, and the number of generated routes is shown in Figure 11. When the metric of \( T_d \) is increasing, both the length and the number of generated routes are increasing as well. The reason is that if \( T_d \) is
getting bigger, more TB patterns will be recognized as a same frequent TB pattern and lead to generating longer and bigger count of routes. When $T_d$ increases, however, the time precision of the candidate routes is declined. For example, assume that the visit duration at a spot is 21 min. If $T_d$ is set as 5 min, then the discrete time is integer 5; the time error is 4 min at the route recommendation phase. If $T_d$ is set as 20 min, then the discrete time is integer 2; the time error increases to 19 min. Further, the accumulating time error of the candidate travel route is unacceptable under an improper $T_d$ value. Therefore, to balance the relation between the quality of TB sequential patterns and the time error of the travel route, $T_d$ is suggested at 10 min in this work.

Regarding the popularity normalization coefficient, we test the coefficient ranging from 4 to 10 segments with 2,000 randomly selected sequences. Figures 12 and 13 illustrate the quality of the generated travel routes with different normalization coefficients. As shown in Figures 12 and 13, when the coefficient increases, the length and the number of generated routes both decrease. This is because that the larger coefficient is set, the more TB patterns can be generated in a travel behavior sequence. That is, more normalized popularity values will decrease the support count of the corresponding TB pattern. However, more normalized popularity values can describe a tourist's preference more precisely, which can enhance the personalization of the recommendation. Therefore, to make a proper balance between the quality and the personalization of the generated travel routes, setting the normalization coefficient as 5 is appropriate for this work. Finally, the observations of Figures 11 and 13 demonstrate that the proposed algorithm is effective in generating diverse travel routes.

5. Conclusions

The main goal of our work is to design a travel route recommendation system that recommends personalized tangible travel routes for various tourists within a given POI. First of all, we designed a novel method based on smartphone and IoT infrastructure to collect onsite travel behaviors of tourists in a specific POI automatically. To learn tourists' preferences to each interesting object or spot, we developed an Android App to record multiple onsite travel behaviors on each spot including visit duration, taking pictures, and standing. Next, we designed a travel behavior sequence preprocessing method and a Tourist-Behavior sequential route mining algorithm to generate potential frequent tangible travel routes. Furthermore, the route ranking method uses the querying tourist's personal profile and route constraint to recommend personalized tangible travel routes. Finally, experimental results demonstrate that the proposed system is efficient and effective in recommending tangible travel routes based on collected onsite travel behavior data.

Our future works include (1) deploying our system in a real-world POI, (2) using more types of smartphone sensors to gather more types of onsite travel behaviors to learn tourists' preference precisely, (3) harnessing real-time congestion information at each spot of a scenic area to generate more reasonable travel routes and further improve tourists' travel experience, and (4) using the current location and historical travel sequence of the querying tourist to generate a real-time route recommendation when the tourist requests route recommendations at an arbitrary location within a POI for improving the flexibility of our system.
Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. U1501252, 61572146, and U1711263), the Natural Science Foundation of Guangxi Province (nos. 2016GXNSFDA380006 and AC16380122), the Guangxi Innovation-Driven Development Project (no. AA17202024), and the Guangxi Universities Young and Middle-aged Teacher Basic Ability Enhancement Project (no. 2018KY0203).

References


