

Article

Next Place Prediction Based on Spatiotemporal Pattern Mining of Mobile Device Logs

Sungjun Lee ¹, Junseok Lim ¹, Jonghun Park ^{1,*} and Kwanho Kim ²

¹ Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Korea; zaregn81@snu.ac.kr (S.L.); kasia05@snu.ac.kr (J.L.)

² Department of Industrial & Management Engineering, Incheon National University, 119 Academy-ro, Yeonsu-gu, Incheon 22012, Korea; khokim@inu.ac.kr

* Correspondence: jonghun@snu.ac.kr; Tel.: +82-2-880-7174

Academic Editor: Leonhard M. Reindl

Received: 1 October 2015; Accepted: 20 January 2016; Published: 23 January 2016

Abstract: Due to the recent explosive growth of location-aware services based on mobile devices, predicting the next places of a user is of increasing importance to enable proactive information services. In this paper, we introduce a data-driven framework that aims to predict the user's next places using his/her past visiting patterns analyzed from mobile device logs. Specifically, the notion of the spatiotemporal-periodic (STP) pattern is proposed to capture the visits with spatiotemporal periodicity by focusing on a detail level of location for each individual. Subsequently, we present algorithms that extract the STP patterns from a user's past visiting behaviors and predict the next places based on the patterns. The experiment results obtained by using a real-world dataset show that the proposed methods are more effective in predicting the user's next places than the previous approaches considered in most cases.

Keywords: next place prediction; movement patterns; spatiotemporal patterns; Markov chain; gapped sequence mining

1. Introduction

Owing to the recent exponential growth of location-aware services based on mobile devices, such as smart phones, smart watches and tablet PCs, predicting a user's next place becomes an important research topic in both academia and industry [1–5]. This problem concentrates on predicting a place that will be visited by a user in advance before she/he arrives, on the basis of the user's past visiting behaviors inferred through utilizing sensors, such as Global Positioning System (GPS) and wireless fidelity (WiFi) sensor, that are commonly available in modern mobile devices.

When the level of geographical granularity for prediction comes into consideration, a more precise level is desired to enable further sophisticated services. Through discovering the next places at the level of users' daily lives, such as local shops and school cafeterias, various customized applications can be enabled, including recommendation of tailored information, such as automated reservation and personalized advertisements [6–8].

To predict a user's next place, three types of patterns, namely sequential, temporal sequential and periodic patterns, have been intensively studied. Mobile sequential patterns have been utilized to predict next places based on frequently-observed sequential patterns of the places visited [9]. Gambs *et al.* proposed a modified version of a Markov chain to predict next places by analyzing mobile movement behaviors [10]. Similarly, Alavarez-Garcia *et al.* and Jeung *et al.* employed a hidden Markov chain-based method to infer a user's final place [11,12].

Gidófalvi *et al.* extended the Markov chain-based approach for predicting next places in a continuous manner by adopting an inhomogeneous continuous-time Markov model [13,14].

Rodriguez-Carrion *et al.* suggested a light version of the Lempel-Ziv (LZ) based prediction algorithm to perform predictions on mobile devices [15,16]. Morzy and Pei *et al.* proposed a rule-based approach that discovers associations between an individual user and a place through utilizing a modified *a priori* algorithm [17–19].

Furthermore, several attempts have been made to enhance the performances of next place prediction by considering both spatial and temporal aspects. Giannotti *et al.* classified users' moving intentions into geographically-triggered and temporally-triggered intentions in terms of place and time [20]. Lu *et al.* proposed a methodology for mining two types of trajectory patterns, periodic behavior and swarm pattern [21]. By using a nonlinear time series analysis, Scellato *et al.* attempted to additionally consider arrival time in predicting next places [22]. Wang and Prabhala proposed a user-specific periodicity model based on each user's visiting history [23].

While the previous work mainly focused on the problem of predicting the user's next locations in terms of cell IDs [15,16,24] or at the levels of intra-city or inter-cities [9,10,14,22,25,26], the geographical granularity considered in this paper is at the level of people's daily lives (e.g., buildings). When applied to such a fine level of location granularity, the previous approaches suffer from one or more limitations due to the following unique characteristics of the next place prediction problem discussed in this paper.

First, mobile device logs are likely to contain much noise and missing data related to users' past visits, which is caused by various reasons, such as measurement errors, wireless connection problems or unpowered mobile devices. Dealing with such noise and missing data is crucial, since they make it difficult to achieve accurate parameter estimation and rule generation, leading to unrealistic predictions of next places eventually. Moreover, as the geographical granularity becomes finer, the impact of such error-prone data on the performance of a prediction method becomes more severe.

Next, compared to the cases with coarse location granularity, there is a larger amount of irregular visits in the past history of a user in the considered problem, which makes it even more difficult for a prediction model to accurately identify the user's visiting patterns. For instance, various irregular visits, such as going shopping or to movies, are frequently found between the regular visits of going to and returning from work. As a result, if those irregular visits are simply ignored, prediction models often fail to capture the patterns hidden among them.

Finally, it is necessary to be able to predict the next place for a user by utilizing only a small amount of observations available for the user, since collecting a user's mobile device log is usually a time-consuming and costly task. Accordingly, the methods, such as rule mining and decision tree, that require a significant amount of history information for prediction do not appear to be a viable option when time and cost are an issue.

Motivated by the above remarks, we attempt to develop a novel framework that aims to predict a user's next place based on the user's past visiting behaviors through considering periodicity in addition to time and location. To address the three challenges mentioned above, the proposed framework maps the individual visit of a user to one of the visiting patterns by utilizing the pattern extraction algorithms and the pattern similarity function proposed in this research.

The proposed framework constructs spatiotemporal (ST) trajectories, each of which represents a sequence of stays in terms of place and time, from a limited amount of past visit data for each user. Spatiotemporal-periodic (STP) patterns are then extracted from the user's ST trajectories by the proposed STP extraction algorithm. The algorithm searches for STP patterns through considering both occurrence frequencies and associations with ST trajectories with respect to time for effective recognition of irregular or new visits as STP patterns. In particular, we employ a smoothing function to deal with the noisy and missing data.

Subsequently, STP trajectories are built by mapping each ST trajectory to an STP pattern that is most similar to the trajectory among the extracted STP patterns. Basing on gapped sequence mining [27], the proposed framework is able to identify user's sporadic visits in her/his daily life through constructing gapped STP (GSTP) trajectories that allow gaps to accommodate irregular visits

that cannot be specified in advance. The next place visited by a user is then predicted by the proposed prediction algorithm based on the user's current and recent visits.

This paper is organized as follows: In Section 2, the details of the proposed methods are described. In Section 3, the data collection details and experimentation results are described. The conclusions are presented in Section 4.

2. Proposed Framework

In this section, we describe in detail how the proposed framework extracts GSTP trajectories and predicts the user's next places through considering sequential, temporal and periodic characteristics of a mobile device log. Figure 1 illustrates the overall training process of the proposed framework that consists of four steps to generate GSTP trajectories. The training process proceeds as follows.

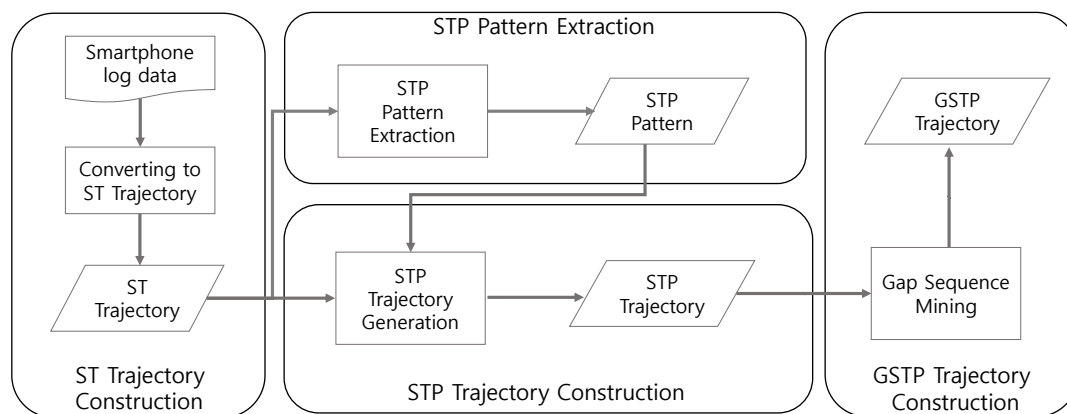


Figure 1. The training process of the proposed approach for computing gapped spatiotemporal-periodic (GSTP) trajectories.

First, an ST trajectory, defined as a sequence of stays in which each stay is represented in terms of a place visited, and the arrival and departure time, as well as the day of week for the visit, is constructed from raw data. Second, we extract STP patterns from ST trajectories to capture periodic revisits by taking periodicity into consideration. The existence of an STP pattern for a user indicates that the user tends to periodically revisit a particular place at a specific time associated with the pattern.

Next, in the STP trajectory construction step of Figure 1, ST trajectories are mapped into a sequence of the extracted STP patterns, named the STP trajectory, based on the similarity between an STP pattern and an element of an ST trajectory. Finally, gap-constrained sequential pattern mining is applied to the STP trajectories to construct a user's GSTP trajectory that allows unobserved places in the user's STP trajectories. The generated GSTP trajectories from the training process are then used for prediction of the next place when the user's most recent STP trajectory data are provided as test data. The detailed descriptions are presented in the following sections.

2.1. WiFi-Based Place Identification

We employ a WiFi fingerprint-based localization method [28] for extracting the places visited from a user's mobile device log. It is well known that this method has advantages over GPS-based approaches when tracking and identifying people's movements in indoor environments, particularly in urban areas, and the method also provides several benefits in terms of energy efficiency, compared to the GPS-based ones, as it utilizes WiFi sensor data.

The localization method requires a WiFi fingerprinting database, containing WiFi access point (AP) data, each of which consists of a place, p , the basic service set identifier (BSSID) and a range of

received signal strengths (RSSI) observed at p . The database is used to infer a user's visit to a place by matching WiFi APs with those in the database according to BSSIDs and their RSSI ranges.

Table 1 shows an example for the WiFi fingerprint-based localization method. The example shows a WiFi fingerprinting database, raw WiFi data and the localization result in Table 1a–c, respectively, where distinct places are indexed. As an example, for an instance observed at 15:00 on 11 November 2013 in Table 1b, the place is identified as p_2 , since only p_2 in the database has the matching BSSIDs and RSSI values in range with those of the instance.

Table 1. Examples of: (a) the WiFi fingerprinting database; (b) raw WiFi data; and (c) places identified by applying a fingerprinting-based localization method. BSSID, basic service set identifier.

(a)					
Place	BSSID	RSSI	Place	BSSID	RSSI
p_1	BSSID_4	−48 to −40	p_2	BSSID_3	−33 to −23
p_1	BSSID_5	−39 to −31	p_3	BSSID_1	−28 to −18
p_2	BSSID_1	−57 to −49	p_3	BSSID_4	−72 to −61
p_2	BSSID_2	−40 to −29	p_3	BSSID_6	−63 to −53
(b)					
Timestamp		BSSID	RSSI		
11 November 2013 15:00		BSSID_1	−55		
		BSSID_2	−34		
		BSSID_3	−22		
11 November 2013 15:45		BSSID_4	−44		
		BSSID_5	−33		
11 November 2013 15:55		BSSID_1	−49		
		BSSID_2	−38		
		BSSID_3	−26		
...					
(c)					
Timestamp			Place		
11 November 2013 15:00			p_2		
11 November 2013 15:45			p_1		
11 November 2013 15:55			p_2		
11 November 2013 16:25			p_2		
11 November 2013 17:15			p_3		
11 November 2013 19:00			p_3		
11 November 2013 19:15			p_2		
11 November 2013 20:00			p_2		
...					

More specifically, the observation of BSSID_1 at 15:00 on 11 November 2013 in Table 1b indicates that the place can be either p_2 or p_3 according to Table 1a, but only the RSSI range of p_2 contains −55, the observed RSSI value. Similarly, the observations of BSSID_2 and BSSID_3 at 15:00 on 11 November 2013 also indicates that place must be p_2 , and therefore, the place at 15:00 on 11 November 2013 is inferred to be p_2 . Through repeating this process for all of the instances in raw WiFi data collected, the places visited by a user, as well as the corresponding timestamps can be constructed. An example of the localization result is shown in Table 1c, where each timestamp is interpreted as the time when the user arrived at a place, indicating that the user was at p_2 from 15:00 to 15:45 on 11 November 2013, for instance.

2.2. ST Trajectory Construction

Once the places and their associated timestamps are identified, the ST trajectory of the i -th day, T_i , is constructed. T_i denotes a sequence of stays, and each stay, $T_{i,j}$, is defined as a four-tuple, (p, t_s, t_f, d) , where t_s and t_f , respectively, represent the start and finish time of the stay in minutes, and $d \in \{Mo, Tu, We, Th, Fr, Sa, Su\}$ is the day of the week for T_i . $T_{i,j}$'s are ordered chronologically in T_i .

$T_{i,j}$ is identified from the log containing the localization result by grouping consecutive logs corresponding to the same place while preserving the ascending order of the timestamps. t_s and t_f are determined to be the start and the finish time of the group, respectively, and p is set to the place of the group. Subsequently, T_i is constructed from identified $T_{i,j}$'s according to the value of i of $T_{i,j}$.

Table 2 shows an example of ST trajectories constructed from the data in Table 1c, assuming that 11 November 2013 is Monday. The first stay, $T_{1,1}$, shown in Table 2, corresponds to the first instance of Table 1c, since the places of the first and second instance of Table 1c are different. As a result, t_s and t_f for $T_{1,1}$ are set to the timestamps of the first and second instance of Table 1c, respectively, leading to $T_{1,1} = (900, 945, p_2, Mo)$. Since the minutes are measured from the beginning of a day, the user is inferred to have stayed at p_2 between 15:00 to 15:45 on Monday. The other stays are generated similarly from Table 1c, and T_i 's are constructed as shown in Table 2.

Table 2. ST trajectory examples.

T_i	$T_{i,j}$	(t_s, t_f, p, d)	T_i	$T_{i,j}$	(t_s, t_f, p, d)
T_1	$T_{1,1}$	$(900, 945, p_2, Mo)$	T_{15}	$T_{15,1}$	$(895, 1080, p_2, Mo)$
	$T_{1,2}$	$(945, 955, p_1, Mo)$		$T_{15,2}$	$(1110, 1125, p_1, Mo)$
	$T_{1,3}$	$(955, 1035, p_2, Mo)$		$T_{15,3}$	$(1165, 1230, p_2, Mo)$
	$T_{1,4}$	$(1035, 1155, p_3, Mo)$		$T_{15,4}$	$(1240, 1320, p_4, Mo)$
	$T_{1,5}$	$(1155, 1260, p_2, Mo)$...
T_8		...			
	$T_{8,1}$	$(900, 1140, p_4, Mo)$			
	$T_{8,2}$	$(1140, 1280, p_3, Mo)$			
		...			

2.3. STP Pattern Extraction

In this paper, we consider a weekly periodicity for extracting STP patterns of a user, as most people have weekly visiting patterns [29]. Extraction of the STP patterns from ST trajectories consists of three steps: grouping ST trajectories based on weekly periodicities, computing the probabilities of a stay according to its periodicity group membership and generating STP patterns from the probabilities.

Specifically, ST trajectories are grouped according to the day of the week contained in T_i to accommodate the weekly periodicity of user's movements. For each group of ST trajectories, the probability of a stay is computed through examining whether or not a user has visited a place at a specific time based on the arrival and departure time. Then, STP patterns are extracted by finding time segments that exceed a certain threshold in terms of the probability. Detailed description of each step is given in the following.

2.3.1. ST Trajectory Grouping for Periodicity Identification

In order to take the periodicity into account, ST trajectories are grouped based on a weekly periodicity, denoted as $D \in \mathcal{D}$, where \mathcal{D} denotes a set of all possible combinations of the days of the week, $\{Mo, Tu, We, Th, Fr, Sa, Su\}$. For instance, $D = \{Mo, We\}$ represents a periodicity of visits that tend to be made on every Monday and Wednesday. We let \mathcal{T}_D be the set of T_i 's containing $T_{i,j}$ whose d belongs to D . For instance, if $D = \{Mo\}$, $\mathcal{T}_{\{Mo\}}$ formed from Table 2 can be expressed as $\{T_i | i = 7(x - 1) + 1, x \in \mathbb{N}\}$, where \mathbb{N} is the set of positive integers.

2.3.2. Computing the Probability of a Stay

Given D , \mathcal{T}_D is used for computing the probability of a stay at place p at a discrete time, $t \in \{1, \dots, t_{max}\}$, denoted as $q_{D,p,t}$, where t and t_{max} , respectively, are the time since the beginning of a day and the time at the end of the day. Both t and t_{max} are measured in minutes, ranging from 1 to 1440, and accordingly $t_{max} = 1440$.

A procedure for calculating $q_{D,p,t}$ based on counting the number of stays in the ST trajectories that belong to \mathcal{T}_D is shown in Algorithm 1 in which $P(\mathcal{T}_D)$ is a function that returns the set of places included in the ST trajectories contained in \mathcal{T}_D . In Line 2, the temporary variable, $Q_{D,p,t}(T_i)$, that stores the probability of a stay for T_i is initialized to zero for all t . In Lines 5 to 11, $Q_{D,p,t}(T_i)$ becomes one if there exists a stay $T_{i,j} = (p, t_s, t_f, d) \in T_i$, such that $t_s \leq t \leq t_f$, and set to the maximum between its current value and the result of smoothing, otherwise. Finally, $q_{D,p,t}$ is computed by averaging $Q_{D,p,t}(T_i)$ across T_i 's in \mathcal{T}_D , in Line 15.

Linear smoothing is applied to $q_{D,p,t}$ to accommodate the variability of a stay, as well as noise in the raw WiFi data. λ_s is a slope parameter that determines the penalty amount for the stays that are not exactly matched in terms of time. The penalty is proportional to the distance between t and t_s or between t and t_f , increasing $Q_{D,p,t}(T_i)$ as t goes near t_s or t_f , but only up to $Q_{D,p,t_s}(T_i)$ or $Q_{D,p,t_f}(T_i)$, respectively.

Algorithm 1: Algorithm for Calculating the Probability of a Stay.

Input: $\mathcal{T}_D, p \in P(\mathcal{T}_D), t \in \{1, \dots, t_{max}\}$
Output: $\{q_{D,p,t}\}$

```

1 foreach  $T_i \in \mathcal{T}_D$  do
2    $Q_{D,p,t}(T_i) \leftarrow 0; \forall t \in \{1, \dots, t_{max}\}$ 
3   foreach  $T_{i,j} = (p, t_s, t_f, d) \in T_i$  do
4     for  $t \leftarrow 1$  to  $t_{max}$  do
5       if  $t < t_s$  then
6          $Q_{D,p,t}(T_i) \leftarrow \max(Q_{D,p,t}(T_i), 1 - \lambda_s(t_s - t))$ 
7       else if  $t_s \leq t \leq t_f$  then
8          $Q_{D,p,t}(T_i) \leftarrow 1$ 
9       else
10         $Q_{D,p,t}(T_i) \leftarrow \max(Q_{D,p,t}(T_i), 1 - \lambda_s(t - t_f))$ 
11      end
12    end
13  end
14 end
15  $q_{D,p,t} \leftarrow \sum_{T_i \in \mathcal{T}_D} \frac{Q_{D,p,t}(T_i)}{|\mathcal{T}_D|}$ 

```

Figure 2a,b respectively shows the calculation results of $Q_{\{Mo\},p_2,t}(T_1)$ and $Q_{\{Mo\},p_2,t}(T_{15})$ over time, based on Table 2 when $D = \{Mo\}$, $\mathcal{T}_D = \{T_1, T_8, T_{15}\}$, and $\lambda_s = 0.05$. Note that $Q_{\{Mo\},p_2,t}(T_8) = 0$ for all t , since T_8 does not include any stay at p_2 , although $T_8 \in \mathcal{T}_D$. Figure 2c shows the plot for $q_{\{Mo\},p_2,t}$ based on $T_{1,1}$, $T_{1,3}$, $T_{1,5}$, $T_{15,1}$ and $T_{15,3}$ for each t , where trapezoid shapes are attributed to the application of smoothing to the stay probabilities.

Specifically, the value of $q_{\{Mo\},p_2,t}$ is computed as follows: First, we consider $T_{1,1}$ and compute $Q_{\{Mo\},p_2,t}(T_1)$. Since t_s and t_e for $T_{1,1}$ are 900 and 945, respectively, $Q_{\{Mo\},p_2,t}(T_1) = 1$ when $t \in [900, 945]$. For other t 's, the linear smoothing function in Algorithm 1 is applied. As a result, $Q_{\{Mo\},p_2,t}(T_1) = 0$ when $t < 880$ or $t > 965$, $Q_{\{Mo\},p_2,t}(T_1) = 1 - 0.05(900 - t)$ when $t \in [880, 900]$ and $Q_{\{Mo\},p_2,t}(T_1) = 1 - 0.05(t - 945)$ when $t \in [945, 965]$. The probabilities of stays at p_2 of $T_{1,3}$ and $T_{1,5}$ are computed similarly, and the results are shown in Figure 2a. Subsequently, we repeat the above probability calculations for all $T_i \in \mathcal{T}_D$ and set $q_{\{Mo\},p_2,t}$ as the average of probabilities computed for T_1 and T_{15} , resulting in the thick line in Figure 2c.

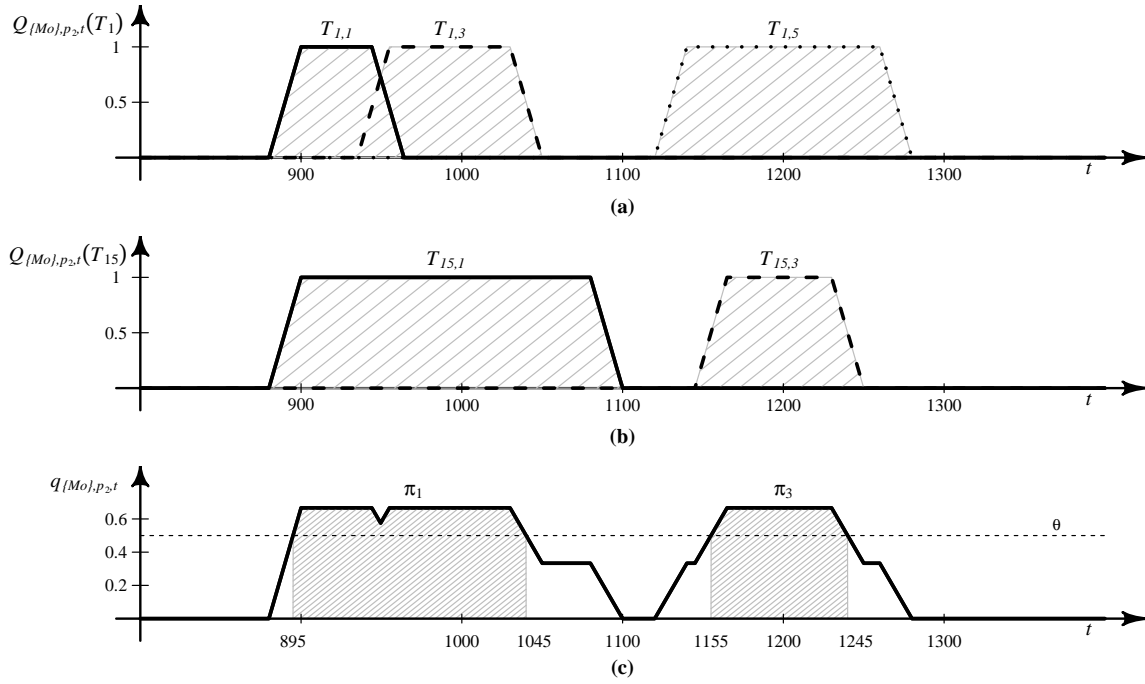


Figure 2. Examples illustrating the calculation of: (a) $Q_{\{Mo\},p_2,t}(T_1)$; (b) $Q_{\{Mo\},p_2,t}(T_{15})$; and (c) $q_{\{Mo\},p_2,t}$.

2.3.3. Extracting STP Patterns

After computing $q_{D,p,t}$ for all D, p and t , we proceed to compute the set of STP patterns for D , denoted as Π_D . We define each STP pattern $\pi \in \Pi_D$ as a triplet, (p, τ_s, τ_f) , where τ_s and τ_f , respectively, stand for the start and finish time of π . Given periodicity D and place p , STP patterns are extracted from $q_{D,p,t}$ by finding the time segments whose associated probabilities are greater than a certain threshold, θ . Detailed descriptions on how to extract the STP patterns are shown in Algorithm 2.

Algorithm 2: Algorithm for STP Pattern Extraction.

Input: $\theta, q_{D,p,t}$; $p \in P(\mathcal{T}_D), t \in \{1, \dots, t_{max}\}$

Output: Π_D

```

1  $E \leftarrow \phi$ 
2 foreach  $p \in P(\mathcal{T}_D)$  do
3   for  $t \leftarrow 1$  to  $t_{max} - 1$  do
4     if  $q_{D,p,t} \geq \theta$  then
5        $E = E \cup \{t\}$ 
6     else
7       if  $E \neq \phi$  then
8          $\tau_s \leftarrow \min\{E\}$ 
9          $\tau_f \leftarrow \max\{E\} + 1$ 
10         $\pi \leftarrow (p, \tau_s, \tau_f)$ 
11         $\Pi_D \leftarrow \Pi_D \cup \{\pi\}$ 
12         $E \leftarrow \phi$ 
13      end
14    end
15  end
16 end

```

In Lines 3 to 15 of Algorithm 2, STP patterns are identified only for the consecutive time epochs whose probabilities are greater than or equal to θ . E is a temporary variable that records the set of consecutive time epochs, such that $q_{D,p,t} \geq \theta$. When $q_{D,p,t}$ falls below θ and E is not empty, a new STP pattern is identified by respectively setting τ_s and τ_f to be the start and finish time of the new pattern in Lines 8 to 9. Then, the new pattern is added to Π_D . The above procedure is repeated for all $p \in P(\mathcal{T}_D)$, and the algorithm finally returns the set of extracted STP patterns, Π_D .

Table 3a shows an example of STP patterns extracted from the ST trajectories in Table 2 when $\mathcal{T}_{\{Mo\}} = \{T_1, T_8, T_{15}\}$ and $\theta = 0.5$. In Table 3a, π_1 is extracted from $T_{1,1}$, $T_{1,3}$ and $T_{15,1}$, while π_3 is from $T_{1,5}$ and $T_{15,3}$. STP patterns, π_1 and π_3 , are illustrated as shaded areas in Figure 2c. For instance, π_1 is extracted from $q_{\{Mo\},p_2,t}$ as follows: the goal is to find the consecutive time segments that satisfy $q_{\{Mo\},p_2,t} \geq \theta$. Since $q_{\{Mo\},p_2,t} = \frac{2 \times (1 - 0.05(900 - t))}{3}$ and $q_{\{Mo\},p_2,t} = \frac{2 - 0.05(t - 1035)}{3}$ when $t \in [880, 900]$ and $t \in [1035, 1055]$, respectively, $t \in [895, 1045]$ satisfies $q_{\{Mo\},p_2,t} \geq \theta$. Therefore, τ_s and τ_f of π_1 are set to 895 and 1045, respectively, and as a result, $\pi_1 = (p_2, 895, 1045)$.

Table 3. Examples of (a) STP patterns; and (b) STP trajectories.

(a)	
π	(τ_s, τ_f, p)
π_1	$(895, 1045, p_2)$
π_2	$(1015, 1175, p_3)$
π_3	$(1155, 1245, p_2)$
...	
(b)	
s	Sequence
s_1	$\langle e_s, \pi_1, \pi_2, \pi_3, e_f \rangle$
s_2	$\langle e_s, \pi_2, e(p_1), e_f \rangle$
s_3	$\langle e_s, \pi_1, \pi_3, e(p_4), e_f \rangle$
...	

Note that the existence of $T_{1,2}$ between $T_{1,1}$ and $T_{1,3}$ was ignored during the construction of π_1 . This is due to the smoothing applied to the probability of a stay, $q_{\{Mo\},p_2,t}$, after the finish time of $T_{1,1}$ and before the start time of $T_{1,3}$, resulting in the effect of treating the user's stay at p_1 during 10 min as a temporary visit that is often observed while a user is moving to another location. Indeed, the smoothing allows us to effectively combine the multiple re-visits to the same place even though the time intervals of their stays do not overlap, while providing a means to deal with temporary or irregular visits.

2.4. Generation of STP Trajectories

The user's movement pattern is represented as an STP trajectory that is generated from ST trajectories by utilizing the extracted STP patterns for the user. We let s denote an STP trajectory. s is a sequence consisting of symbols, each of which corresponds to an STP pattern or event. It starts with event e_s and ends with event e_f , respectively indicating the start and finish of s . The set of s 's generated for weekly periodicity D is denoted as S_D .

s is constructed by replacing each stay in an ST trajectory with the STP pattern that is most similar to the ST trajectory while sequentially exploring each ST trajectory in the ascending order of time. The similarity between a stay and an STP pattern is calculated based on overlap between the time segments of the stay and the pattern.

Specifically, the similarity between $T_{i,*}$ and π^* , denoted as $t\text{-sim}(T_{i,*}, \pi^*)$, is defined as Equation (1).

$$t\text{-sim}(T_{i,*}, \pi^*) = \frac{\text{length of overlap between time intervals of } T_{i,*} \text{ and } \pi^*}{\text{length of time interval of } T_{i,*}} \quad (1)$$

Algorithm 3 shows the detailed procedure for generating a set of STP trajectories from \mathcal{T}_D and Π_D , given a threshold for pattern similarity, θ' . In Algorithm 3, $\langle \cdot \rangle$ is used for representing a sequence, and \oplus denotes an operator for the concatenation of two sequences.

Algorithm 3: Algorithm for STP Trajectory Construction.

Input: \mathcal{T}_D, Π_D
Output: S_D

```

1  foreach  $T_i \in \mathcal{T}_D$  do
2     $s \leftarrow \langle e_s \rangle$ 
3    while  $T_i \neq \phi$  do
4       $\Pi'_D \leftarrow \Pi_D$ 
5       $s^* \leftarrow \phi$ 
6       $T_{i,*} = T_{i,j}$  with the smallest  $t_s$ ;  $\forall T_{i,j} = (p, t_s, t_f, d) \in T_i$ 
7       $T_i \leftarrow T_i / \{T_{i,*}\}$ 
8      while  $\Pi'_D \neq \phi$  do
9         $\pi^* = \pi$  with the smallest  $\tau_s$ ;  $\forall \pi = (p, \tau_s, \tau_f) \in \Pi'_D$ 
10        $\Pi'_D \leftarrow \Pi'_D / \{\pi^*\}$ 
11       if  $t\text{-sim}(T_{i,*}, \pi^*) \geq \theta'$  then
12          $s^* \leftarrow s^* \oplus \langle \pi^* \rangle$ 
13       end
14     end
15     if  $s^* = \phi$  then
16        $p^* \leftarrow P(T_{i,*})$ 
17        $s^* \leftarrow \langle e(p^*) \rangle$ 
18     end
19      $s \leftarrow s \oplus s^*$ 
20   end
21    $s \leftarrow s \oplus \langle e_f \rangle$ 
22    $S_D \leftarrow S_D \cup \{s\}$ 
23 end

```

In Line 2, e_s is added to s to represent the start of s , and in Line 6, stay $T_{i,*}$ with the smallest t_s is selected to traverse in the ascending order of time and is removed from T_i in the next line. From Line 8 to 14, the algorithm attempts to find the matching patterns for $T_{i,*}$ based on $t\text{-sim}()$ among the candidate STP patterns by traversing the patterns in Π'_D one by one in the chronological order of τ_s .

When there is no matching pattern found for $T_{i,*}$, the event of visiting place $P(T_{i,*})$, denoted as $e(p^*)$, instead of an STP pattern is added to s as in Lines 15 to 18, where $P()$ is a function that returns the place contained in stay $T_{i,*}$. The STP trajectory for T_i is augmented with s^* in Line 19, and e_f is appended to s to indicate the end of the sequence in Line 21. Finally, the constructed STP trajectory s is added to S_D as a member, and the algorithm returns S_D as an output.

Table 3b shows an example of STP trajectory construction result from the ST trajectories in Table 2 by applying Algorithm 3 with the STP patterns defined in Table 3a. In Table 3b, STP trajectory s_1 consists of three STP patterns, π_1 , π_2 and π_3 . For constructing s_1 , π_1 is selected as a matching pattern for $T_{1,1}$, since π_1 has the highest similarity among the STP patterns considered. Actually, $t\text{-sim}(T_{1,1}, \pi_1)$ was one as the time interval of $T_{1,1}$ is included in that of π_1 . For s_3 , an event of visiting place p_4 rather than an STP pattern is inserted at the fourth position, since there exists no matching pattern related to visiting p_4 around that time.

2.5. Gapped Sequence Mining

Among many sequential pattern mining algorithms that have been proposed in the past to discover frequent patterns from sequences, the gapped sequence mining algorithm has been known to provide satisfactory results in many applications [30]. It extracts patterns with consideration of gap constraints when finding frequent subsequences to relax the consecutiveness requirement on the subsequences. We employ a gap-constrained sequential pattern mining algorithm, known as cSPADE (Sequential Pattern Discovery using Equivalence classes with constraints) [27], to discover frequent subsequences from STP trajectories. It allows us to deal with irregular visits, as well as uncertainties in a mobile device log due to the presence of noisy data by using gap symbols.

Table 4a presents the result of applying the cSPADE algorithm to the STP trajectories in Table 3b. The outputs of the cSPADE algorithm are frequent subsequences with gaps, as well as their confidence values, which are then used to generate GSTP trajectories. The confidence of a sequence indicates the likelihood of occurrence of the last symbol in the sequence, given that all of the preceding symbols before the last one have been observed.

Table 4. Examples of the: (a) results of the cSPADE algorithm, as well as their confidences; and (b) GSTP trajectories.

(a)		
No.	Subsequence with Gap	Confidence
1	$\langle e_s, \pi_1 \rangle$	0.84
2	$\langle \pi_3, e_f \rangle$	0.67
3	$\langle \pi_1, \pi_3, e_f \rangle$	0.75
4	$\langle \pi_1, \pi_3, e(p_4) \rangle$	0.78
5	$\langle \pi_1, S_g, \pi_3, S_g, e_f \rangle$	0.49
6	$\langle e_s, \pi_1, \pi_2, \pi_3, e_f \rangle$	0.55
7	$\langle e_s, \pi_1, \pi_3, e(p_4), e_f \rangle$	0.27
...		
(b)		
σ	(s^p, s^c, s^s, u)	
σ_1	$(\phi, e_s, \pi_1, 1.44)$	
σ_2	$(\phi, \pi_3, e_f, 1.27)$	
σ_3	$(\langle \pi_1 \rangle, \pi_3, e_f, 1.25)$	
σ_4	$(\langle \pi_1 \rangle, \pi_3, e(p_4), 1.28)$	
σ_5	$(\langle \pi_1, S_g \rangle, \langle \pi_3, S_g \rangle, e_f, 0.89)$	
σ_6	$(\langle e_s, \pi_1, \pi_2 \rangle, \pi_3, e_f, 1.15)$	
σ_7	$(\langle e_s, \pi_1, \pi_3 \rangle, e(p_4), e_f, 0.87)$	
...		

For instance, the fifth sequence in Table 4a is obtained from s_1 and s_3 of Table 3b, where S_g denotes a set of sequences, each of which is an empty set or a sequence composed of events or patterns for representing a sequence of gap symbols. That is, subsequence of s_1 , $\langle \pi_1, \pi_2, \pi_3, e_f \rangle$, can be obtained from $\langle \pi_1, S_g, \pi_3, S_g, e_f \rangle$ by substituting the first S_g with π_2 and the second S_g with ϕ . Similarly, the subsequence of s_3 , $\langle \pi_1, \pi_3, e_f \rangle$, can be generated by replacing all of the S_g 's with ϕ .

More formally, a GSTP trajectory, defined as four-tuple $\sigma = (s^p, s^c, s^s, u)$, is obtained from a frequent subsequence in such a way that s^s and s^c , respectively, are the symbols at the last and the second to last positions of the frequent subsequence, and s^p corresponds to the rest. That is, a frequent subsequence found by cSPADE is split into three parts that respectively represent the current STP pattern or event (*i.e.*, s^c), the preceding patterns or events (*i.e.*, s^p) before s^c and the succeeding pattern or event (*i.e.*, s^s) after s^c .

When S_g is located at the second to last position in a frequent subsequence, its immediate predecessor together with S_g form s^c , and all of the other predecessors constitute s^p . cSPADE is applied for each weekly periodicity D in \mathcal{D} , and the resulting GSTP trajectories are stored into Σ_D .

Once GSTP trajectories are obtained, the average length of S_g contained in s^p of σ can be computed by counting the number of symbols corresponding to S_g for each STP trajectory used to discover σ during the training process and taking their average. For instance, since σ_5 of Table 4b has been derived from s_1 and s_3 of Table 3b and $S_g = \{\pi_2\}$ for s_1 and $S_g = \phi$ for s_3 , the average length of S_g for s^p of σ_5 is $(1 + 0)/2 = 0.5$. The average length of S_g in s^c can be computed in the same way.

Finally, $u(\sigma)$ represents the utility of GSTP trajectory, σ , when making a prediction of the next place, and it is defined as Equation (2).

$$u(\sigma) = (\text{confidence of } \langle s^p, s^c, s^s \rangle) + \lambda(1 + \text{average length of } S_g \text{ in } s^p)^{-1} + \lambda'(1 + \text{average length of } S_g \text{ in } s^c)^{-1} \quad (2)$$

where λ and λ' are weight parameters.

$u(\sigma)$ considers not only the confidence of a frequent subsequence, but also the average length of gaps located in s^p and s^c to accommodate the uncertainty associated with a GSTP trajectory. Note that the utility of a GSTP trajectory decreases as the gap symbols become longer. Furthermore, we set λ' to be greater than λ to put more emphasis on the utility related to the current and next places.

Table 4b presents an example of GSTP trajectories generated from the frequent subsequences in Table 4a, where λ and λ' were set to 0.1 and 0.5, respectively. There is a one-to-one correspondence between the subsequence of Table 4a and the GSTP trajectory of Table 4b. As an example, we consider the fifth subsequence in Table 4a, which is $\langle \pi_1, S_g, \pi_3, S_g, e_f \rangle$. s^s , s^c and s^p of σ_5 are e_f , $\langle \pi_3, S_g \rangle$ and $\langle \pi_1, S_g \rangle$, respectively, as S_g is at the second to last position in the subsequence. Therefore, $u(\sigma_5)$ is $0.49 + 0.1 \times \frac{1}{1+0.5} + 0.5 \times \frac{1}{1+0.5} = 0.89$.

2.6. Next Place Prediction

Figure 3 depicts the test process for predicting the next location of a user when a new observation on the user's movement is made. In order to predict the next place, it is necessary to convert the user's movement logs to an STP trajectory and then to compare it to the GSTP trajectories identified during the training process. The steps involved in the test process are exactly the same as those in the training process in Figure 1, except for skipping the STP pattern extraction step for generating an STP trajectory. Once an STP trajectory is obtained from the test data, the user's next place is predicted by Algorithm 4, which finds the most similar GSTP trajectory to the STP trajectory and predicts the next location by following the GSTP trajectory found.

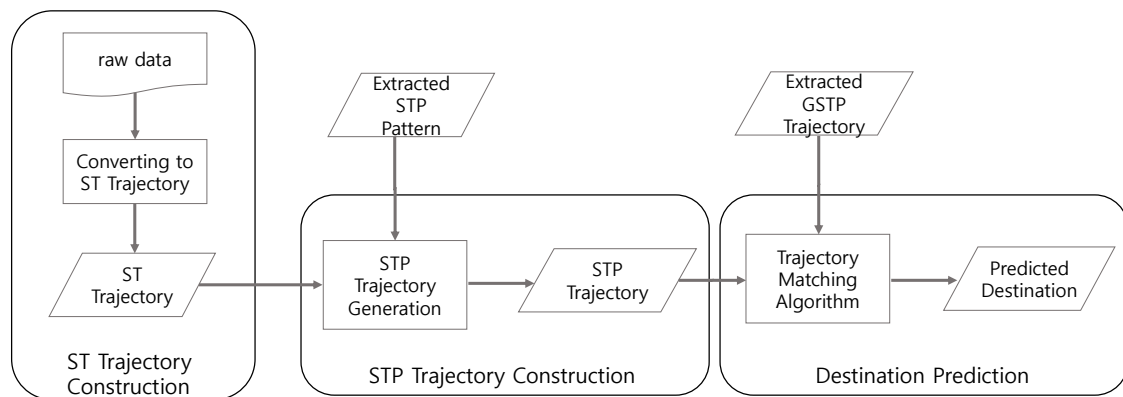


Figure 3. The test process for predicting next places based on raw test data.

Algorithm 4: Algorithm for Next Location Prediction.**Input:** Σ_D, s_A **Output:** p^*

- 1 s_A^c = symbol at the last position of s_A .
- 2 s_A^p = sequence preceding s_A^c in s_A .
- 3 Find $\sigma = (s^p, s^c, s^s, u)$ with the highest $p\text{-sim}(s^p, s_A^p)$ s.t. $s^c \ni s_A^c; \forall \sigma \in \Sigma_D$
- 4 The ties are broken by picking σ with the highest u .
- 5 $p^* \leftarrow$ place p of s^s

Algorithm 4 describes how the proposed framework infers the next place from an STP trajectory of user A , based on input data s_A , which is the STP trajectory of user A given as test data and Σ_D , GSTP trajectories constructed during the training process. s_A is split into s_A^c and s_A^p , which respectively are the last symbol that can be either an STP pattern or an event corresponding to the currently visiting place, and all of the symbols preceding s_A , denoting the past movements.

In Line 3 of Algorithm 4, the best matching STP pattern σ is found by examining the entire GSTP trajectories Σ_D . $\sigma = (s^p, s^c, s^s, u)$ is obtained by use of the similarity between two sequences, $p\text{-sim}(s^p, s_A^p)$, that measures the length of overlapping subsequences between the sequences corresponding to the past movements. The similarity function is defined in Equation (3) in which S_g is counted as of a length of one when calculating the length.

$$p\text{-sim}(s^p, s_A^p) = \frac{\text{length of the intersection between } s^p \text{ and } s_A^p}{\text{length of } s_A^p} \quad (3)$$

When there exist more than one STP pattern with same similarity value, the tie is broken by picking σ with the highest u . The algorithm then returns p^* , which is the place contained in s^s representing a pattern or an event. Finally, execution of Algorithm 4 is repeated for all $D \in \mathcal{D}$ to select the best σ across the various weekly periodicities.

As an example, we assume $s_A = \langle \pi_1, \pi_3 \rangle$, implying that user A is currently at p_2 (from Table 3a). Among the GSTP trajectories in Table 4b, σ_3 and σ_4 have the highest similarity, 1.0, as $s_A^p = \langle \pi_1 \rangle$ is the same as s^p 's of σ_3 and σ_4 . Between σ_3 and σ_4 , we choose σ_4 , since $u(\sigma_4) > u(\sigma_3)$. Accordingly, p_4 is predicted to be the user A 's next place, as $e(p_4)$ of s^s of σ_4 indicates an event of visiting place p_4 .

3. Experiments

3.1. Dataset

Among several types of mobile devices, we adopted smartphones as data collection devices, since they are equipped with WiFi sensors and frequently carried by users anywhere they go throughout their daily activities. For experimentation, we implemented an Android mobile app that records the data pertaining to user's visits, such as timestamps and WiFi signals, every minute. The mobile app was then distributed to eight students at Seoul National University (SNU), and the data were collected during two months spanning from September to November 2013.

The subjects were chosen in such a way that they have different majors; half of them are residents of a campus dormitory; and half of them take classes for more than 4 days a week, so that they can represent different campus lifestyles. As all of the participants were undergraduate students and the experiments were conducted during a semester, most activities they performed during the study period were related to typical campus life, including having a meal at a cafeteria, taking a class in a classroom, sleeping in a dormitory, doing homework in the library and doing exercise at a gym.

Since our research was a part of a smart campus project that aims to study intelligent services facilitating better campus life, data collection was conducted only inside the SNU campus, and all of the places considered were located within the campus. Another reason for limiting the scope to the SNU campus only was due to the availability of the WiFi fingerprinting database required by the

proposed approach. Building a WiFi fingerprinting database involves time-consuming tasks and is very costly, but only the database for the SNU campus was available at the time of this research.

Throughout the experiments, all participants were instructed to carry their mobile devices with them as much as possible to gather comprehensive data that can reflect their actual daily movements. The full dataset contains 714,448 WiFi signal logs, and 19.85 WiFi APs were detected on average for each observation. Since the logs also include locations outside campus, only about 52 percent of logs were successfully mapped into meaningful places based on a localization method using the WiFi fingerprinting database for campus buildings. Furthermore, the first 42 days' logs out of 60 days were selected as training data for constructing the prediction model, and the rest was used for evaluating the model's performance.

Figure 4 shows an example of a subject's ST trajectories retrieved from the collected data. In this figure, blocks of the same gray level indicate visits to the same place, and white backgrounds represent unknown locations. The horizontal axis corresponds to the time from 0:00 to 24:00 of a day, while the vertical axis represents the number of days from the beginning of the experimentation. That is, the horizontal block stands for the subject's stay at some place from the time at which the block begins until the time at which the block ends, and appearances of the blocks with the same gray level along with the vertical axis indicate that the subject visited the same place at similar time slots across the days.

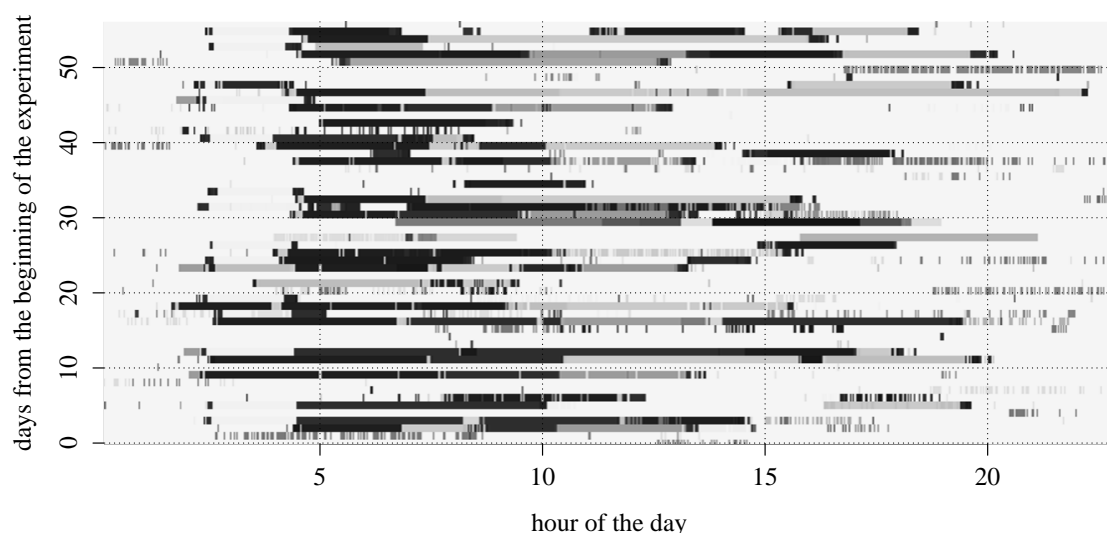


Figure 4. Visualization of a subject's ST trajectories where blocks of the same gray level indicate the visits to the same place and white backgrounds represent unknown locations.

From Figure 4, it can be observed that frequent revisits to the same place were usually made with weekly periodicities rather than daily due to the characteristics of campus life, and accordingly, we have extracted patterns based on the weekly periodicity. Yet, there are many irregular or exceptional visits that can be attributed to noisy observations, errors during localization or participant's peculiarities, making the problem of next place prediction difficult. We address this difficulty by use of smoothing for constructing STP patterns and also by applying gapped sequence mining during the generation of GSTP trajectories.

3.2. Parameter Settings

For STP pattern extraction and STP trajectory construction, the parameters were determined experimentally by taking the values that maximize the performance of the proposed model. Figure 5a,b shows the prediction accuracy results when varying θ , θ' and λ_s individually while the other parameters were fixed. From Figure 5a,b, it can be seen that large θ hurts the performance as more false

STP patterns are introduced, and roughly 50 min of smoothing are appropriate for identifying a stay. The highest performance was achieved when we respectively set θ , θ' and λ_s to 0.06, 0.16 and 0.02.

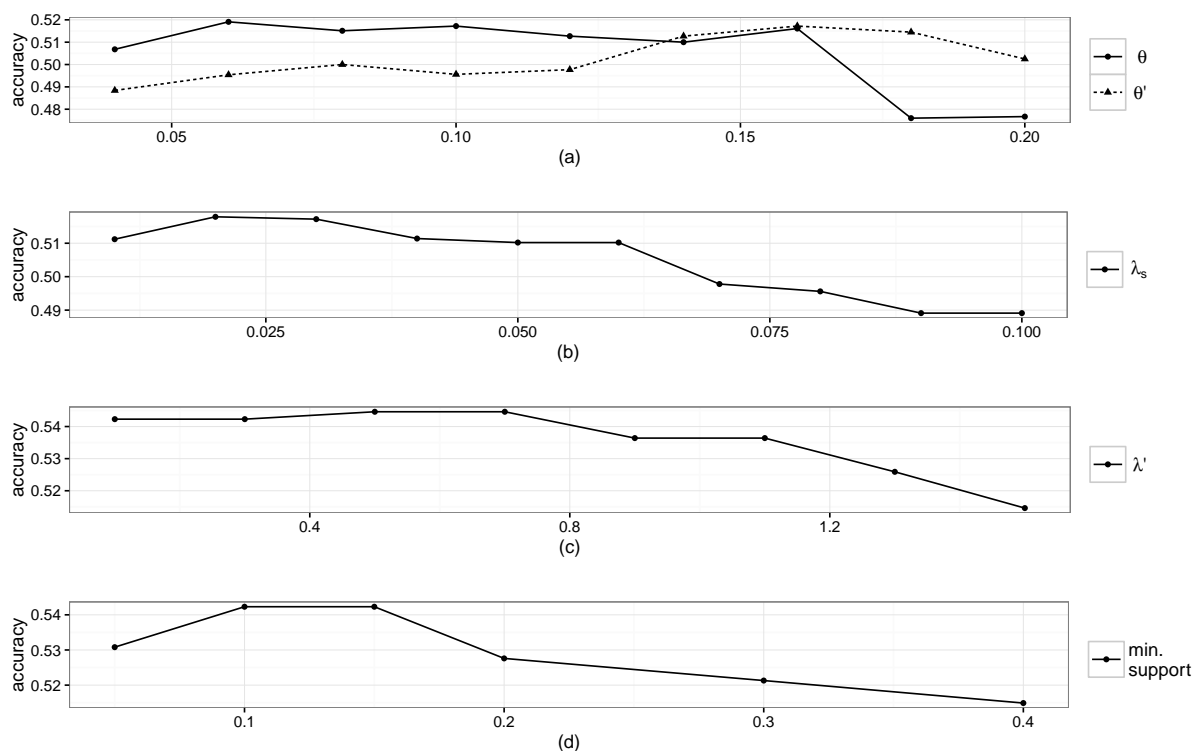


Figure 5. Accuracy results when varying parameters: (a) θ and θ' ; (b) λ_s ; (c) λ' ; and (d) minimum support.

The maximum gap, maximum window size, minimum support, λ and λ' are the parameters involved in the gapped sequence mining. Individual effects of λ' and the minimum support on the accuracy are respectively plotted in Figure 5c,d, where the maximum performance was achieved when setting λ' and the minimum support to 0.5 and 0.15, respectively. Performance differences were negligible when varying the values of the maximum gap, maximum window size and λ , and we set them to 3, 7 and 0.1, respectively.

Finally, several weekly periodicities were selected in consideration of the characteristics of campus life, which include the periodicities based on a single day, except Saturday and Sunday, and those based on typical class schedules at SNU, resulting in $\mathcal{D} = \{\{Mo\}, \{Tu\}, \{We\}, \{Th\}, \{Fr\}, \{Mo, We\}, \{Tu, Th\}, \{Mo, We, Fr\}\}$.

3.3. Evaluation Results

In order to demonstrate the effectiveness of the proposed framework, we have implemented two first-order Markov chain-based methods that predict the next place by calculating the probabilities for all of the possible next places based on the transition probabilities among places and choosing the place with the highest probability. We remark that the same ST trajectory data (like those in Table 2) were used for both the proposed methods and the first-order Markov chain methods to be fair with the presence of noisy data in the comparison.

The comparison results for the proposed methods and the Markov chain methods in terms of the accuracy metric are presented in Figures 6 and 7, in which MC, MC-P, STP and GSTP, respectively stand for (1) the Markov chain method without periodicity consideration; (2) the Markov chain method with periodicity consideration; (3) the prediction based on STP trajectories; and (4) the prediction based on GSTP trajectories.

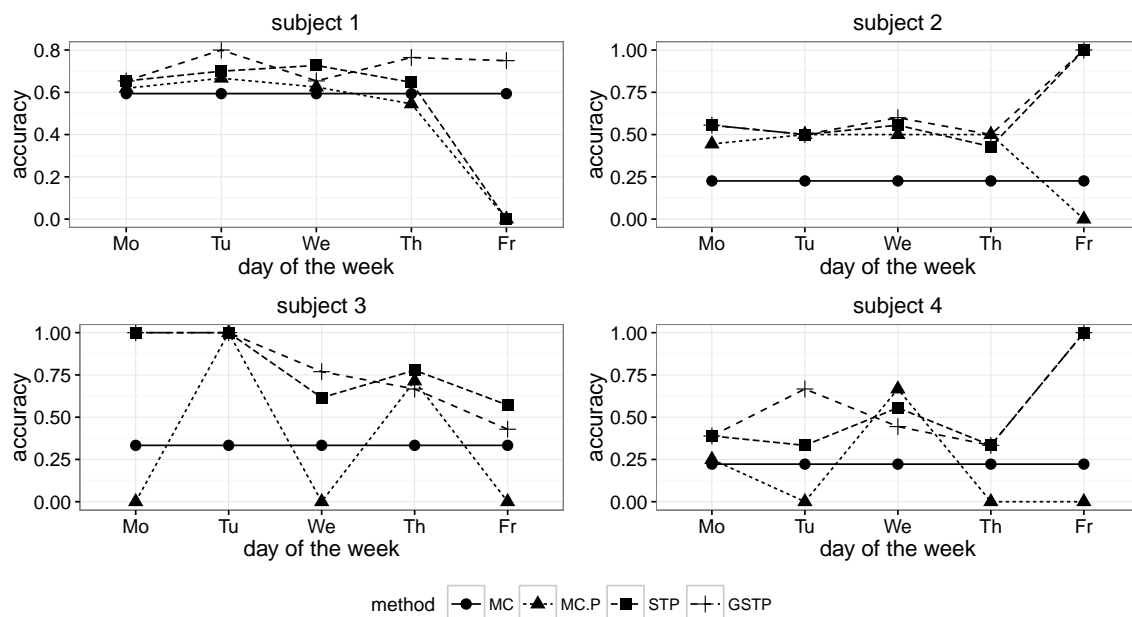


Figure 6. Accuracy results across the days of the week for Subjects 1 to 4. MC, Markov chain; P, periodicity.

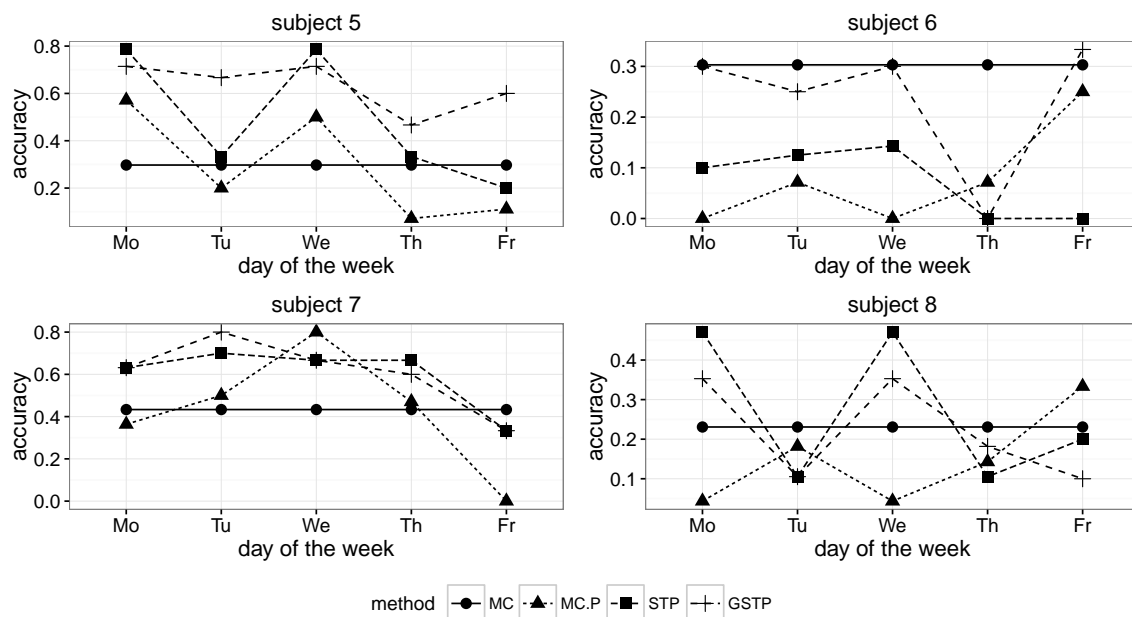


Figure 7. Accuracy results across the days of the week for Subjects 5 to 8.

While MC predicted the next locations by using all of the available ST trajectory data without taking the day of week information into account, MC-P exploited the day of week information by selectively utilizing ST trajectories grouped by weekly periodicities according to the day on which prediction was made. Since MC prediction was performed on all of the trajectories in the training data, its accuracy results are the same across the day of the week, as shown in Figures 6 and 7.

On the other hand, STP is based on the STP trajectory data (e.g., Table 3b) for the prediction that was made by choosing the pattern or event that has the highest transition probability from a current pattern or event after computing the transition probabilities between patterns or events. Finally, GSTP trajectories (e.g., Table 4b) were used for predicting the next location with the GSTP method.

As shown in Figure 8, the overall accuracy results of MC and MC-P were worse than those of STP and GSTP. These poor performances yielded by the Markov chain-based methods are due to their inability to address the irregularities of visits, which is the characteristic often observed in campus life. In particular, the performance results of MC-P imply that the periodicity alone cannot help with increasing the accuracy. Figure 8 also shows that GSTP slightly outperformed STP on average, while their performance variabilities barely differ. Furthermore, it can be observed from Figure 8 together with Figures 6 and 7 that STP and GSTP tend to provide more stable performances across the different days of the week than MC-P.

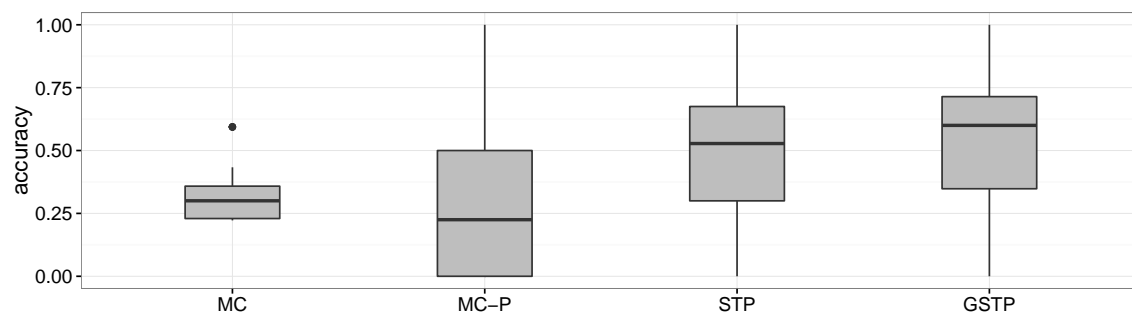


Figure 8. Boxplots for the accuracy results of the methods compared.

The next places were far from being predictable for some subjects, owing to the high irregularity in visiting behaviors when the MC and MC-P were used, but the prediction performances were greatly improved for them when applying the proposed methods, STP and GSTP. In particular, STP and GSTP significantly outperformed MC and MC-P for Subjects 3–5, as shown in Figures 6 and 7.

Accordingly, it appears that the proposed notion of STP trajectory facilitates accuracy enhancement through generalizing observations into patterns, as well as accommodating periodicities. In addition, incorporation of gaps into the pattern sequence by the GSTP method was also successful for further increasing the accuracy. These together imply that the proposed framework was effective at predicting the user's next location.

3.4. Effects of Movement Regularity

Besides the overall accuracy, we found out that the performance of the proposed methods significantly varied depending on the lifestyle of a subject. After the data collection experiment, we conducted a short survey asking about the regularity assessment for the subject's movements during the study period in terms of a 3-point Likert scale. A score of 3 was reported by Subjects 1–3, indicating that they managed highly regular life patterns. On the other hand, the score of Subject 5 was 1, whereas the score of the rest was 2.

Based on this survey result, it appears that the performance of the proposed method was satisfactory when a subject exhibited highly regular behaviors, leading to the average prediction accuracy of more than 0.7 for Subjects 1 and 3. In contrast, when the visiting behavior of a subject was not very regular, the prediction performances of STP and GSTP were low, as suggested by the results for Subjects 6 and 8.

To further explore the relationship between the regularity of movements and the prediction performance, we computed Jaccard similarity [31], which measures overlaps among the visited places by a subject for each day of the week, and employed it as a metric for assessing the regularity. Figure 9 shows the result that contains 40 plots corresponding to five different days of the week for eight subjects and their resulting performances.

The regularity score varied according to the subject, as well as the day of the week. The Pearson correlation coefficient for the plots in Figure 9 was 0.267, indicating a weak positive relationship between the regularity and the prediction accuracy of GSTP, which suggests that the regularity

alone cannot fully explain the prediction performance due to the GSTP's ability of accommodating irregularities through the smoothing and gapped sequence mining. It is still interesting to note that we can observe more dots in the upper right corner of Figure 9 for Subjects 1 to 3, who reported the highest scores for their subjective regularity assessment than for the other subjects, and *vice versa*.

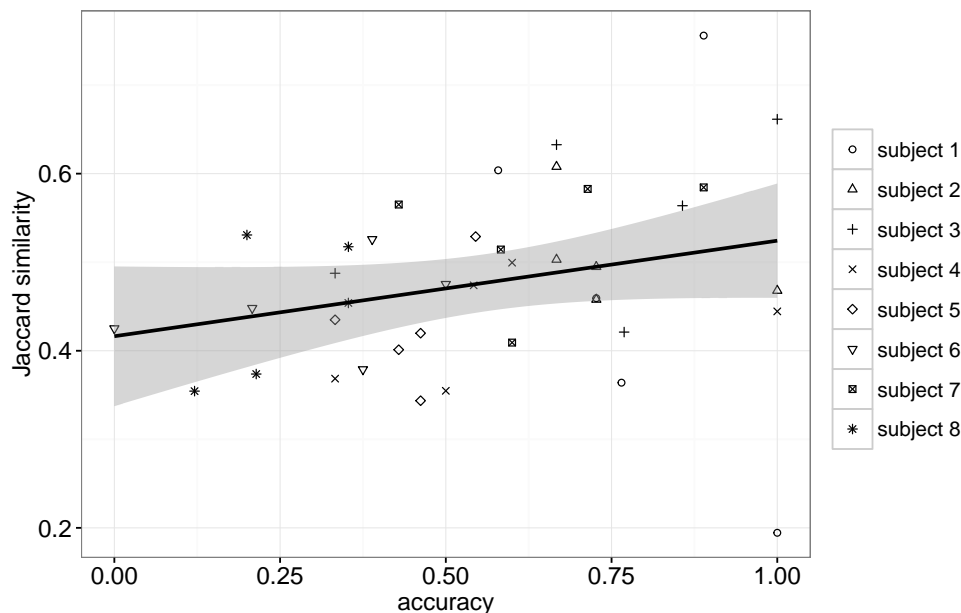


Figure 9. Scatter plot between the Jaccard similarity and the accuracy of GSTP.

4. Conclusions

In this paper, we exploited time, location and periodicity information to effectively predict the user's next place through introducing the notion of the STP pattern and the application of gapped sequence mining. Frequently- and periodically-observed visiting behaviors were recognized as STP patterns for a user, and the patterns were then used for representing the user's past visits as STP trajectories. Subsequently, the extracted STP trajectories were further generalized to GSTP trajectories to accommodate irregularities of visits, as well as to deal with exceptional stays.

Through the experimentation based on a real-world dataset collected from eight people, it was found that the proposed methods outperform the conventional methods based on the Markov chain in terms of prediction accuracy.

As future work, we plan to apply our work to larger and more complex environments than a university campus, such as urban areas or travel sites with more participants, and to further enhance the proposed spatiotemporal-periodic patterns through developing more sophisticated similarity measures that can effectively accommodate diverse types of irregularities and the semantic meaning of places.

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIP) (No. 2013R1A2A2A03013947).

Author Contributions: Sungjun Lee conceived of the idea, developed the proposed methods and wrote the paper. Sungjun Lee and Junseok Lim performed the experiments. Jonghun Park advised the research and revised the paper. Kwanho Kim critically reviewed and revised the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, M.; Hsu, W.J. Mining GPS data for mobility patterns: A survey. *Pervasive Mob. Comput.* **2014**, *12*, 1–16.
2. Rashidi, P.; Cook, D.J. Mining and monitoring patterns of daily routines for assisted living in real world settings. In Proceedings of the 1st ACM International Health Informatics Symposium, Arlington, VA, USA, 11–12 November 2010; pp. 336–345.
3. Sadilek, A.; Krumm, J. Far Out: Predicting Long-Term Human Mobility. In Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; pp. 814–820.
4. Monreale, A.; Pinelli, F.; Trasarti, R.; Giannotti, F. Wherenext: A location predictor on trajectory pattern mining. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 637–646.
5. Xia, H.; Qiao, Y.; Jian, J.; Chang, Y. Using smart phone sensors to detect transportation modes. *Sensors* **2014**, *14*, 20843–20865.
6. Ashbrook, D.; Starner, T. Using GPS to learn significant locations and predict movement across multiple users. *Pers. Ubiquit. Comput.* **2003**, *7*, 275–286.
7. Amini, S.; Brush, A.; Krumm, J.; Teevan, J.; Karlson, A. Trajectory-aware mobile search. In Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, Austin, TX, USA, 5–10 May 2012; pp. 2561–2564.
8. Kim, B.; Kang, S.; Ha, J.Y.; Song, J. VisitSense: Sensing place visit patterns from ambient radio on smartphones for targeted mobile ads in shopping malls. *Sensors* **2015**, *15*, 17274–17299.
9. Ying, J.J.C.; Lee, W.C.; Tseng, V.S. Mining Geographic-temporal-semantic Patterns in Trajectories for Location Prediction. *ACM Trans. Intell. Syst. Technol.* **2014**, *5*, doi:10.1145/2542182.2542184.
10. Gambs, S.; Killijian, M.O.; del Prado Cortez, M.N. Next place prediction using mobility markov chains. In Proceedings of the First Workshop on Measurement, Privacy, and Mobility, Cambridge, UK, 10 April 2012; p. 3.
11. Alvarez-Garcia, J.; Ortega, J.; Gonzalez-Abril, L.; Velasco, F. Trip destination prediction based on past GPS log using a Hidden Markov Model. *Expert Syst. Appl.* **2010**, *37*, 8166–8171.
12. Jeung, H.; Shen, H.T.; Zhou, X. Mining Trajectory Patterns Using Hidden Markov Models. In *Data Warehousing and Knowledge Discovery*; Song, I.Y., Eder, J., Nguyen, T.M., Eds.; Springer: Berlin, Germany, 2007; pp. 470–480.
13. Gidófalvi, G.; Pedersen, T.B. Mining long, sharable patterns in trajectories of moving objects. *Geoinformatica* **2009**, *13*, 27–55.
14. Gidófalvi, G.; Dong, F. When and where next: Individual mobility prediction. In Proceedings of the 1st ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; pp. 57–64.
15. Rodriguez-Carrion, A.; Garcia-Rubio, C.; Campo, C. Performance evaluation of LZ-based location prediction algorithms in cellular networks. *IEEE Commun. Lett.* **2010**, *14*, 707–709.
16. Rodriguez-Carrion, A.; Garcia-Rubio, C.; Campo, C.; Cortés-Martín, A.; Garcia-Lozano, E.; Noriega-Vivas, P. Study of LZ-based location prediction and its application to transportation recommender systems. *Sensors* **2012**, *12*, 7496–7517.
17. Morzy, M. Prediction of Moving Object Location Based on Frequent Trajectories. In *Computer and Information Sciences—ISCIS 2006*; Levi, A., Savas, E., Yenigun, H., Balci, S., Saygin, Y., Eds.; Springer: Berlin, Germany, 2006; pp. 583–592.
18. Morzy, M. Mining Frequent Trajectories of Moving Objects for Location Prediction. In *Machine Learning and Data Mining in Pattern Recognition*; Perner, P., Ed.; Springer: Berlin, Germany, 2007; pp. 667–680.
19. Pei, J.; Han, J.; Mortazavi-Asl, B.; Pinto, H.; Chen, Q.; Dayal, U.; Hsu, M.C. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proceedings of the 29th IEEE International Conference on Data Engineering, IEEE Computer Society, Heidelberg, Germany, 2–6 April 2001; pp. 215–224.
20. Giannotti, F.; Nanni, M.; Pinelli, F.; Pedreschi, D. Trajectory Pattern Mining. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 12–15 August 2007; pp. 330–339.
21. Lu, E.C.; Tseng, V.; Yu, P. Mining Cluster-Based Temporal Mobile Sequential Patterns in Location-Based Service Environments. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 914–927.

22. Scellato, S.; Musolesi, M.; Mascolo, C.; Latora, V.; Campbell, A.T. NextPlace: A spatio-temporal prediction framework for pervasive systems. In *Pervasive Computing*; Springer: Heidelberg, Germany, 2011; pp. 152–169.
23. Wang, J.; Prabhala, B. Periodicity based next place prediction. In Proceedings of the Nokia Mobile Data Challenge 2012 Workshop, Newcastle, UK, 18–19 June 2012; Volume 2.
24. Mathew, W.; Raposo, R.; Martins, B. Predicting future locations with hidden Markov models. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 911–918.
25. Tran, L.H.; Catasta, M.; McDowell, L.K.; Aberer, K. Next place prediction using mobile data. In Proceedings of the Mobile Data Challenge Workshop (MDC 2012), Newcastle, UK, 18–19 June 2012.
26. Yang, J.; Xu, J.; Xu, M.; Zheng, N.; Chen, Y. Predicting next location using a variable order Markov model. In Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming, Dallas, TX, USA, 4 November 2014; pp. 37–42.
27. Zaki, M.J. Sequence mining in categorical domains: incorporating constraints. In Proceedings of the 9th International Conference on Information and Knowledge Management, McLean, VA, USA, 6–11 November 2000; pp. 422–429.
28. Zhang, P.; Zhao, Q.; Li, Y.; Niu, X.; Zhuang, Y.; Liu, J. Collaborative WiFi Fingerprinting Using Sensor-Based Navigation on Smartphones. *Sensors* **2015**, *15*, 17534–17557.
29. Li, Z.; Han, J. Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data. In *Data Mining and Knowledge Discovery for Big Data*; Springer: Heidelberg, Germany, 2014; pp. 41–81.
30. Li, Y.; Bailey, J.; Kulik, L.; Pei, J. Mining probabilistic frequent spatio-temporal sequential patterns with gap constraints from uncertain databases. In Proceedings of the 2013 IEEE 13th International Conference on Data Mining (ICDM), Dallas, TX, USA, 7–10 December 2013; pp. 448–457.
31. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).