ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

N-Most Interesting Location-based Recommender System

Sumet Darapisut[1] Komate Amphawan[2] Sunisa Rimcharoen[3] and Nutthanon Leelathakul[4]

ABSTRACT

The popular and ubiquitous location-based social networks (LBSNs) appeal many users for sharing interesting locations with other users. As the collected data (such as users' profile, location's comment, and suggestion) become a lot larger in size, location-based recommender systems require more effective filters to be able to suggest potentially preferable locations to users. Location recommendation is more difficult and challenging especially if users have few or no check-in histories as a new user. Therefore, instead of depending on users' check-in histories, previous works focused on creating recommended location lists by leveraging the information given by other users who check in locations in each area. However, previous studies took into account only the frequencies of users for creating recommendation lists and have suffered from the cold-start problem where new users have few or none histories. As a result, the recommenders hardly suggest any locations matching the users' preference. In this paper, we propose an enhanced location-recommendation approach called N-most interesting location-based recommender system (NILR) to recommend interesting locations for new users. Our approach can be divided into three phases. First, the NILR discovers interesting locations by taking into account both the visiting frequencies and the preferences of users already in the system. Second, a ranking procedure is executed to create a final recommendation list based on two interestingness scores: one obtained from the HITS-based model (as adopted by [1] and [2]) and the other from our proposed method. Finally, we re-filter interesting locations based on the current location of the new user. Experimental results reveal the NIRL can reach better precision, recall, average ranking and NDGC than HITS by 6%, 6%, 30% and 8% for Tokyo and 24%, 30%, 43% and 15% for New York dataset, respectively.

Article information:

Keywords: Location-based recommender systems, Non-profile users, Ranking

Article history:
Received: April 26, 2021
Revised: July 2, 2021
Accepted: November 4, 2021
Published: March 12, 2022 (Online)

DOI: 10.37936/ecti-cit.2022161.247546

1. INTRODUCTION

Location-based social networks (LBSNs) have been growing rapidly. This can be seen by counting the number of users and locations. This is possible because of the prevalence of Internet and mobile phone technologies, which facilitate the use of LBSNs [3], [4], [5], [6]. Users in social networks share locations together with corresponding opinions and suggestions for the locations they have visited in the real world. There are many popular LBSN applications such as Foursquare, Facebook Place, and Yelp. They allow users to share check-in data and add new locations. The sets of data (including user profiles, location profiles and relationships between users and locations) collected by the applications become very large, making location recommendation more challenging.

To address the issue just mentioned, location-based recommender systems (LBRSs) [7], [8], [9] play an important role in decision making, by filtering and recommending potentially preferable locations to users. Business owners might rely on LBRS to attract more new customers, resulting in more profit. Most LBRSs perform efficiently in the case of dense user's check-ins or long visiting histories as it is relatively easy to capture users' preference. Nevertheless, in real-world scenarios, users tend to visit locations in their vicinity. They might have few check-ins and are unable to visit all locations, most of which are far

[1,2,3,4]The authors are with Faculty of Informatics, Burapha University, Chonburi, Thailand 20130, E-mail: dearsumet@gmail.com, komate@gmail.com, rsunisa@buu.ac.th and nutthanon@buu.ac.th

away from the current location. Hence, recommending locations to a new user or a traveler from out of town is much more challenging. To cope with such scenarios, several works attempt to discover interesting locations based on visiting habits of local users.

Most existing works discover interesting locations based only on frequencies of check-in or visits [1], [2], [10], in which some locations may have a large number of visits of many users but have return visit counts of 0 (some users visit them only once and never return). In this case, the locations may be newly popular or they may be known as a must to be visited only once in a life time. On the other hand, some interesting but unpopular locations might have a moderate number of visits but users always return more than once. To the best of our knowledge, there exists no work that takes into account visiting frequency, locations' return visits, and each user's return counts to discover interesting locations. Additionally, LBRSs have also suffered from the cold-start problem where new users have no or few check-in history entries causing recommenders to suggest hardly any locations that match the users' preference.

In this work, we present a novel location recommendation approach called N-most interesting location-based recommender system (NILR) to recommend interesting locations for new users (extended from [11]). Our NILR approach can be divided into three phases. First, the NILR can discover interesting locations based on previous users already in the system by taking into account frequency of visiting and preferences of users. The preferences of users can be considered by noting who visits a location more than once. Interesting locations and users have a synergistic relationship. Interesting locations are visited repeatedly and frequently by various users. Very knowledgeable users always visit and revisit numerous interesting locations, possessing more insight and knowledge. Second, to rank locations in the recommendation list, we rank outstanding locations based on interestingness scores, frequencies, and preferences. In other words, users obtain a recommended location list, composed of locations selected effectively by considering diverse aspects instead of only one aspect as in traditional methods. Lastly, when a new user requests a nearby interesting location recommendation, our method filters out locations too far away from the user's current location and, based on the interestingness scores, sorts all locations in descending order. We evaluated our proposed method using a real-world dataset provided by Foursquare. The data set is composed of the check-in's information associated with five areas in Tokyo city: Chiyoda, Minato, Shinjuku, Shibuya and Chuo, and one area in New York city. The data is arranged by the number of locations in decreasing order [12]. For evaluating accuracies of lists recommended for new users, we simulate geospatial ranges, each of which covers all the visited locations in each user's current area, by forming minimum bounding rectangles (MBRs) [1], [13]. The accuracies are measured in terms of precision, recall, ranking measures (*i.e.*, average ranking [14] and normalized discounted cumulative gain, NDCG [15]) and are compared with those of the HITS-based model. Experimental results reveal NILR performs effectively and efficiently, and it outperforms HITS in terms of accuracies and rankings.

The main contributions of our work can be summarized as follows:

- We propose a novel approach which can address the issue of cold start users, named N-most interesting location-based recommender system (NILR), to recommend interesting locations.
- To discover interesting locations, both check-in frequencies and preferences are considered as opposed to other traditional methods, which rely only on frequencies.
- We also propose a ranking algorithm to select outstanding locations for creating recommendation lists based on interestingness scores by taking into account both visitation frequencies to and preferences of users for locations.
- We evaluated our method on a real-world dataset, provided by Foursquare, that is an extensive dataset in LBSNs and we use four metrics including precision, recall, average ranking, and normalized discounted cumulative gain (NDCG).

2. RELATED WORK

This section provides background information and the motivation behind the location-based social network system and reviews existing location-based recommender systems. They can be divided into two categories: location recommendations based on user's experience, and recommendations based on choices made by local experts or other users in the system.

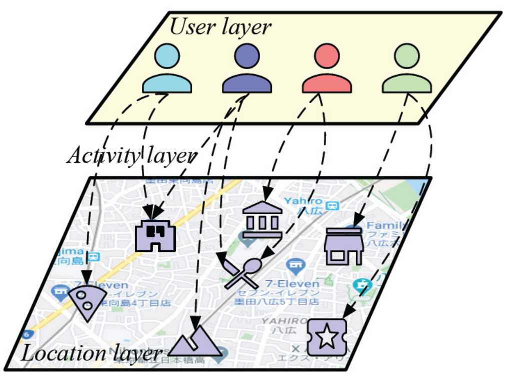

Fig.1: An example of location-based social networks.

2.1 A location-based social network system

Location-based social network systems (LBSNs) [5], [6], [8], [9], have been popular in recent years due to the advances in mobile technologies and the Internet. Users can access LBSNs easily via smartphone applications, which allow users to share their locations. Also, they often share comments or suggestions regarding the interesting locations that they checked in, such as shopping malls, restaurants, coffee shops, bakeries, and convenience stores. According to Foursquare's report in June 2016, there were more than 8 billion check-ins each month at more than 65 million locations with over 55 million users in the world [6]. Yelp disclosed that there were about 29 million active users per month in the system [6]. Thanks to the large number of users and locations contributing to tremendous amount of available data, there are many extensive research works on location-based recommender systems (LBRSs), which is one research area in the field of recommender systems. There are 3 main types of objects to be recommended [9]: users, locations, and activities, as shown in Fig. 1. Users (either travelers or local experts) are the ones who could influence their friends/followers or share similar preferences with others in interesting areas. Their common interests will be used as information to generate location recommendation lists and they can also be used as objects for recommendation [17], [18], [19]. There are two ways of recommending locations that users might prefer: a single location, and a sequence of places to visit, called a trip [20], [21], [22]. The activities are things that users do at a certain time in specific places. Interesting activities are recommended by considering popular activities of the users such as shopping in a mall in the afternoon, and watching drama in a theater at night [12], [23], [24].

Related to our work are location recommendations based on 1) users' experience and 2) local experts or other users in the system.

2.2 Location Recommendations based on users' experience

In 2013, Quan [20] presented a location recommendation algorithm that applied a collaborative filtering approach based on users' behavior. Furthermore, this work took into account the time context of a user. This is called a time-aware POI recommendation, and alleviates a data sparsity problem by considering temporal preference at other time slots of each user in conjunction with finding other users who have similar temporal preferences with a target user (who asks for a recommendation). A category-aware POI recommendation model [25] was proposed to capture user preferences by regarding location categories in transition patterns of a user. A matrix factorization was chosen for creating location recommendation lists. In addition, this method grouped users who have similar check-in time periods and considered the users1 current location. Daniel and Trevor [26] studied users' behaviors and locations in the Foursquare dataset such as frequent users' check-in patterns, time periods of returning to already-visited locations, distribution of checked-in locations, and checked-in location categories over time periods. As a result, this work could recommend potential locations where a user may visit later. In 2017, an adaptive POI recommendation approach, incorporating activities of each user in time context with spatial features called CTF-ARA, was presented [27]. The CTF-ARA method classified users into active and inactive users by using a k-means algorithm. Active users have a large number of check-ins. Time context is regarded for recommending locations to an active user. Inactive users are provided recommendations based on the locations without considering the time context. Jiuxin [28] proposed a location recommendation approach by analyzing users' check-in patterns. The patterns are categorized into three features: fine-grained time intervals, global popularity of locations, and users' personal preferences. Then a classification model is used for predicting the next check-in location of users. Next, a Point-of-Interest (POI) recommendation method was proposed [29]. The method considers three various aspects: 1) location attributes such as latitude, longitude, categories, and user's check-in time, 2) user attributes such as suggestions, reviews, and ratings that users provided, and 3) the other information such as distances of from POIs to user's location and users' social friendship. These aspects were regarded with a ranking-based and a matrix factorization-based method for generating POI recommendation lists. Mingxin and Ling [30] introduced a location recommendation method named a memory-based POI preference attenuation model by taking into account two aspects: user's movements, and a memory-based preference attenuation model by using Ebbinghaus's forgetting curve. These aspects were used to determine the similarity between the target user and others. Then, a collaborative filtering method based on the proposed similarity was used to create recommendation lists. Lastly, a user profile awareness recommendation algorithm (ISC-CF) was proposed [31] to recommend interesting locations and address the sparsity problems by taking into account reviews, user check-in histories, friend relationships, and the users' current location. Latent Dirichlet Allocation (LDA) and collaborative filtering were used to extract interesting locations visited by users and to recommend interesting locations, respectively.

The previous works mentioned above demand data about users' experiences to correctly recommend locations. They yield relatively low accuracy in the case where users are new to the system (i.e., their profile/activity histories have never been collected). In contrast, our work requires no such information.

NILR is based on local experts or other users who are already in the system. Therefore, it is more suitable for recommending locations to users with few check-ins.

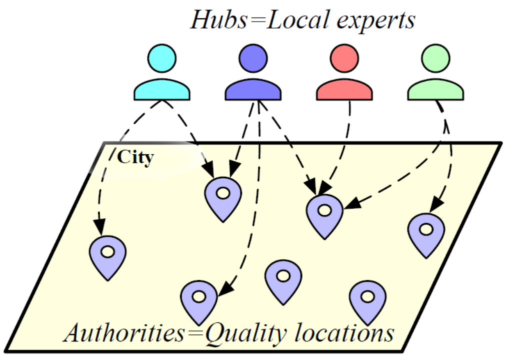

Fig.2: An example of the HITS-based model.

2.3 Location Recommendation based on local experts or other users in the system

Most of the location-based recommendation systems which consider local experts or other users in the system rely on the HITS (Hypertext Induced Topic Search) based model [32] that was originally used for ranking web pages in terms of authority scores and hub scores. An authority is a popular page linked by many hub pages which many authority pages liked. Several works applied this concept by mapping a meaning of the term authorities to locations and hubs to users. Thus, high authority scores mean interesting locations. Highly experienced users (with high hub scores) represent local experts, as shown in Fig. 2. Yu [10] proposed the Tree-Based Hierarchical Graph (TBHG). This work originally proposed the method of modeling travel-sequences of various users on a variety of geospatial sizes and discovering users who have high travelling experience (so called local experts) to recommend interesting locations by using the traditional HITS-based model. Later, Jia [1] presented location-based and preference-aware recommendation systems. The HITS-based model was applied for discovering local experts and then a Weighted Category Hierarchy (WCH) is created for all users. For generating recommendation lists, collaborative filtering (CF) is used for predicting interesting locations based on the similarity between the target user and local experts in the WCH. Xuelian and James [33] proposed a HITS-based POI recommendation algorithm regarding social relationships that focused on relationships among friends more than those among ordinary users. In addition, users, with high experiences from the HITS-based model are discovered by taking into account varieties of visited locations. Likewise, locations for recommending are considered by varieties of local experts measuring in terms of entropy. In 2016, a context-aware location recommendation algorithm (CLoRW) [2] was proposed by using a graph model to represent popular locations, friendship relationships, and local experts. The popular locations based on other users in the system and local experts are figured out with the HITS-based model. Lastly, a random walk algorithm is used for generating location recommendation lists. A personalized successive POI recommendation approach was presented [34]. This work considered three factors to create recommendation list including successive behavior, locality behavior, and group preferences. A distance-weighted HITS algorithm is exploited to discover popular locations based on other users in the system. Later, a location recommendation system based on a Context-aware Tensor Decomposition (CTD) and a weighted HITS were presented [13]. This work suggested locations for an individual effectively. To capture users' preferences, the CTD method took into account temporal influences, including time context and location categories. The weighted HITS algorithm based on user friendships between a target user and other users was applied to generate recommendation lists. In 2019, a dynamic recommender system for suggesting shopping places on Foursquare was proposed [35]. This work searched for local experts in two aspects including experts in their location (called local authorities) and experts in query categories (called topic authorities). They also considered the social contacts of a target user. If the user has no social contacts, a global network graph is created. Otherwise, they created a privacy-aware network graph from the user's social contacts. Then, the network properties of the user are used as guidance to select appropriate algorithms (i.e., PageRank and Lazy random walk algorithms) to find local experts.

The aforementioned research works considered location recommendation lists based on local experts or other users in the system to improve the accuracy of recommendation lists. Nevertheless, several previous works in this field require users' profiles to determine matching local experts or other users in the system. Consequently, they might be unsuitable for new users or travelers without collected profiles. In [33], Long and Joshi consider users based on varieties of locations using entropy measures but this measure has high sensitiveness to check-in data. For example, if a user checks in every location once and another checks in almost every location more than once. The first user has higher entropy values than that of the latter, which may not be correct. In [2], [13], [35], they leverage social relationships to recommend locations. However, friends do not always share common preferences. Collaborative filtering based on users of similar preferences appears to yield more accurate recommendation lists. In [34], the work discovers interesting locations based on distances, which could take more computational time in the online phase to calculate distances between users and locations. In [35], positive reviews are used to filter interesting locations. However, fake reviews might exist, and might cause biased recommendation lists. A lot of reviews on each shopping mall are needed for analyzing sentiment precisely. In addition, not all systems collect or provide reviews. Several previous works in this field require users' profiles for producing accurate recommendation lists, as shown in Table 1. They might be unsuitable for recommending locations to new users or travelers without collected profiles.

In our method, we focus on a location recommendation method based on other users in the system. Location recommendations are different from book or movie recommendations in that users always check in physical locations in their vicinity not far from their office or home, resulting in much sparser data of user experiences or profiles. Therefore, we propose an enhanced location recommendation method based on other users in the system which is more suitable for recommending locations to new users. This is achieved by taking into account visiting frequency, location return visits, and each user's return counts, instead of merely using frequencies. Revisiting represents a strong preference of each user for each location because he or she preferred to use the location's services more than once. Such locations can be implied as interesting locations. However, visiting frequencies of various users are also important to capture popular and interesting locations. We rank outstanding locations based on interesting scores by considering frequency of visiting and users' preferences (i.e., return visits).

3. OUR METHOD

In this section, we first describe basic definitions regarding the N-most interesting location-based recommendation problem. Then, the details of our proposed system, N-most interesting location-based recommender system (NILR), are introduced. The NILR considers a set of locations of food shops including restaurants, cafés, coffee, and desert shops. It measures the interestingness of each location by regarding both visiting frequencies and user preferences for visiting places. Lastly, the N-most interesting locations are recommended to new users unfamiliar with their current location.

3.1 Problem Statement

Let C be a target city which has m food shops (including restaurants, cafés, coffee, and desert shops) of which locations are represented as $L = \{l_1, l_2 \ldots, l_m\}$. The location l_j of the j^{th} shop is associated with its latitude and longitude $< lat_{l_j}, lon_{l_j} >$. The set of shops can be represented as $L = \{< l_1, lat_{l_1}, lon_{l_1} >, < l_2, lat_{l_2}, lon_{l_2} >, \ldots, < l_q, lat_{l_q}, lon_{l_q} >\}$. Subsequently, let $U = \{u_1, u_2, \ldots, u_p\}$ be a set of p users who have visited at least one shop in L. Whenever a user $u_k \in U$ goes to visit a shop $l_j \in L$, his visit is recorded as $v^{u_k,l_j} =< u_k, l_j, f >$, where f denotes a frequency (i.e., the number of times the user u_k has visited the shop l_j). A database $DB = \{v_1, v_2, \ldots, v_z\}$ contains an ordered set of z visiting records of all users in U who visited the shops in L.

The problem of location-based recommendation is to discover a set of N most interesting locations (of shops) for a new user nu_x (where $nu_x \notin U$) who are unfamiliar with their current location, and who ask for a list of recommended places, defined as $RL^{nu_x} = \{rl_1, rl_2, \ldots, rl_N\}$, each of which is an interesting shop $l_j \in L^{nu_x}$ (with a high interestingness score) which the user nu_x should visit.

3.2 The proposed NILR system

Our NILR system takes visiting histories (e.g. check-in, sharing location, etc.) of users (already in the system) as input. Then, it recommends the N-most interesting locations by examining visiting histories of all users. As shown in Fig. 3, the NILR system consists of three main steps: 1) calculation of interestingness of all locations in L, 2) location ranking based on their interestingness, and 3) generation of the top-N most interesting locations for a specific user.

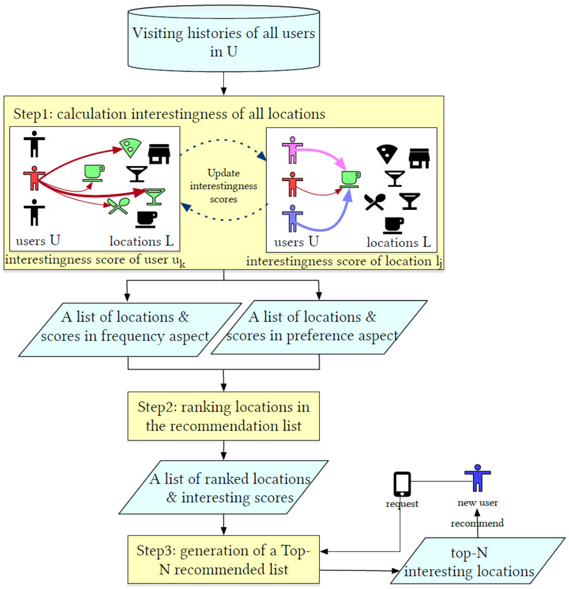

Fig.3: Our method.

3.2.1 Calculation the interestingness of all locations

To compute the interestingness of each location ($l_j \in L$), the visiting history of all users related to l_j is considered iteratively. The interestingness of the location l_j can be expressed by two scores: 1) the one based on its visiting frequency, and 2) the other

Table 1: Features for considering interesting locations based on local experts or other users in the system.

Research works	Features for considering interesting locations						
	Freq.	Variety	Revisit	Review	Distance	Friends	User's profile
[1]	✓	–	–	–	–	–	✓
[2]	✓	–	–	–	–	✓	✓
[10]	✓	–	–	–	–	–	✓
[13]	✓	✓	–	–	–	✓	✓
[33]	✓	✓	–	–	–	✓	✓
[34]	✓	–	–	–	✓	–	✓
[35]	–	–	–	✓	–	✓	–
NILR	✓	–	✓	–	–	–	–

based on visiting preference. These two values can be iteratively computed as described in the following definitions.

Definition 1: An interestingness score of the location l_j based on frequency of visits is calculated by letting $V^{l_j} = \{v_1, v_2, \ldots, v_p\}$ be the ordered set of visiting records of all users at the location l_j. Each visiting record $v_x \in V^{l_j}$ such that $v_x =< u_x, l_j, f_x >$ denotes the number of times, f_x, that the user u_x has visited the location l_j. The interestingness score of l_j based on the frequency of visits is determined by summing each user's visiting score (which is determined by multiplying frequency of visits at l_j by the user's interestingness score) as shown in Equation 1.

$$isf(l_j) = \sum_{x=1}^{p} (f_x \times isf(u_x)) \quad (1)$$

$isf(u_x)$ is u_x's interestingness score (firstly initialized as 1), which is further updated as shown in Eq. 2.

Definition 2: An interestingness score of the user u_k is based on her visiting behavior. It can be computed by letting $V^{u_k} = \{v_1, v_2, \ldots, v_q\}$ be the ordered set of visiting records of the user u_k at any locations in L. Each visiting record ($v_y \in V^{u_k}$) such that $v_y =< u_k, l_y, f_y >$ indicates the number of times, f_y, that the user u_k has visited the location l_y. Then, the interestingness score of the user u_k based on his or her visiting behavior is equal to the summation of each location's visiting score which represents how many times he or she has ever visited (determined by multiplying her frequency of visits at l_y and the interestingness score of l_y), as shown in Equation 2.

$$isf(u_k) = \sum_{y=1}^{q} (f_y \times isf(l_y)) \quad (2)$$

$isf(l_y)$ is l_y's interestingness score as computed by Eq. 1.

However, the values of $isf(l_j)$ and of $isf(u_k)$ grow after each calculation. It is necessary to perform an L2-normalization on the $isf(l_j)$ and $isf(u_k)$ (so that they eventually converge) using Equations 3 and 4.

$$isf(l_j) = \frac{isf(l_j)}{\sqrt{\sum_{y=1}^{|L|} isf(l_y)^2}} \quad (3)$$

$$isf(u_k) = \frac{isf(u_k)}{\sqrt{\sum_{x=1}^{|U|} isf(u_x)^2}} \quad (4)$$

On the other hand, to consider the preference of a user u_x for a location l_j, the number of revisits is considered and calculated. If the user u_x visits the shop l_j more than once, it can be said that the user u_x likes or prefers the shop. The level of a user's preference for a location is calculated as follows:

Definition 3: A preference of a user u_x for a location l_j is identified by either 0 or 1, expressing whether the user u_x repeatedly visits the location l_j, as shown in Equation 5.

$$pref(u_x, l_j) = \begin{cases} 1, & f_x > 1 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

f_x in $< u_x, l_j, f_x >$ is the number of times that the user u_x has visited the location l_j.

Definition 4: A preference score of a user u_x is the number of locations that the user u_x has ever revisited, defined in Equation 6.

$$pref(u_x) = \sum_{j=1}^{|L|} pref(u_x, l_j) \quad (6)$$

Definition 5: An interestingness score of the location l_j based on preference of users can be calculated by letting $V^{l_j} = \{v_1, v_2, \ldots, v_p\}$ be the ordered set of visiting records of all users at the location l_j. Each visiting record ($v_x \in V^{l_j}$) such that $v_x =< u_x, l_j, f_x >$ denotes the number of times, f_x, that the user u_x visits the location l_j. The interestingness score of l_j based on preference of users is the summation of each user's visiting score (which is the product of the user's number of visits at l_j, the user's preference score, and the users interestingness score based on preference), and is defined in Equation 7.

$$isp(l_j) = \sum_{x=1}^{p} (f_x \times pref(u_x) \times isp(u_x)) \quad (7)$$

$isp(u_x)$ is a user score based on his or her preference, firstly initialized as 1 and updated later as explained in Def. 7.

Definition 6: A preference score of a location l_j based on visiting of users is the number of users in U that prefer the location l_j (*i.e.* the users that visit l_j more than once) defined in Equation 8.

$$pref(l_j) = \sum_{x=1}^{|U|} pref(u_x, l_j) \quad (8)$$

Definition 7: An interestingness score of the user u_k based on preference can be calculated by letting $V^{u_k} = \{v_1, v_2, \ldots, v_q\}$ be the ordered set of visiting records of the user u_k at any locations in L. Each visiting record ($v_y \in V^{u_k}$ such that $v_y =< u_k, l_y, f_y >$) identifies the number of times, f_y, that the user u_k visits the location l_y. The interestingness score of the user u_k based on his or her location preference is determined by summing the preference scores of each location he or she has ever visited (which is the product of the user's frequency of visits at the location, the preference score of the location, and the interestingness score of l_y based on users' preference) as shown in Equation 9.

$$isp(u_k) = \sum_{y=1}^{q} (f_y \times pref(l_y) \times isp(l_y)) \quad (9)$$

$isp(l_y)$ is an interestingness score of the location l_y based on the preference of users, determined by Eq. 7.

After each calculation of the interestingness score $isp(l_j)$ for the location l_j and of the interestingness score $isp(u_k)$ for the user u_k, we perform L2-normalization as shown in Equations 10 and 11.

$$isp(l_j) = \frac{isp(l_j)}{\sqrt{\sum_{y=1}^{|L|} isp(l_y)^2}} \quad (10)$$

$$isp(u_k) = \frac{isp(u_k)}{\sqrt{\sum_{x=1}^{|U|} isp(u_x)^2}} \quad (11)$$

As mentioned above, the calculation of the interestingness of all locations in L is iterative. It is necessary to assign a number of iterations (t) to limit the number of computations. All the procedure details are shown in Algorithm 1. The user scores based on frequency and preference of visiting are initialized as 1. Then, for each iteration i of computation ($1 \leq i \leq t$), the interestingness score of each location $l_j \in L$ based on frequency and preference of visiting are first calculated by applying Eq. 1 and Eq. 7. After that, the interestingness score of each user $u_k \in U$ based on his or her frequency and preference of visiting are then calculated. At the end of the process, each location's interestingness score is retained for further computation.

3.2.2 Ranking locations in the recommendation list

To identify interesting locations, the interestingness scores of the locations (based on frequency and preference of visiting calculated from the previous step) are considered. Then, the locations with higher interestingness scores (either on frequency or preference) are ranked with lower numbers. On the other hand, the locations with lower interestingness scores are ranked with higher numbers.

As detailed in the Ranking procedure of Algorithm 1, the ranking number is initialized to be 1. Then, the ranking process is performed repeatedly until all of locations are ranked. First, the maximum value of interestingness score on frequency and preference of visiting, max^{freq} and max^{pref}, are computed. Then, each location l_j having interestingness score $isp(l_j)$ equal to max^{freq} is identified and included into the set L^{freq}. Likewise, each location l_k having $isp(l_k) = max^{pref}$ is included into L^{pref}. Next, the values of max^{freq} and max^{pref} are compared to assign ranking numbers to locations in L^{freq} and L^{pref} in three cases:

1. If $max^{freq} = max^{pref}$, all locations either in L^{freq} or L^{pref} are ranked with the same ranking number.
2. If $max^{freq} < max^{pref}$, all locations in L^{pref} are ranked first and all locations in L^{freq} are ranked later.
3. If $max^{freq} > max^{pref}$, all locations in L^{freq} are ranked first and all locations in L^{pref} are ranked later.

After the ranking, all locations in L^{freq} and L^{pref} are removed from the set of (unranked) locations L and all of their interestingness scores are also removed from ISF^L and ISP^L. The details are shown in the *Ranking* procedure of Algorithm 1.

3.2.3 Generation of a Top-N recommended list

To recommend interesting locations to a new user nu_x, the user needs to give his current location, $loc(nu_x)$. Then, the system selects and ranks the N most interesting locations such that each location is not far from his current location (i.e., each has distance from $loc(nu_x)$) less than the prior-assigned maximum distance, z). The details are shown in the *TopNGeneration* procedure of Algorithm 1.

4. EXPERIMENTAL EVALUATION

In this section, we describe real-world datasets used in this experiment, the experiment settings, and performance measurements. We evaluate the accuracies of the recommendation lists and of the rankings of the top N most interesting locations obtained

Algorithm 1 N-most interesting location-based recommender system

Require: A visiting database, $DB = \{v_1, v_2, \ldots, v_z\}$,
A set of locations, $L = \{l_1, l_2, \ldots, l_q\}$,
A set of users, $U = \{u_1, u_2, \ldots, u_p\}$,
A number of iterations for processing, t
Current location of a new-user nu, $loc(nu)$,
A maximum distance (in km) between current location of the user and a recommended shop, z, and
A number of shops to be recommended, N
Ensure: An order list of N-most interesting locations, $L^{nu} = \{l_y, \ldots, l_z\}$
$ISF^L, ISP^L = \textbf{\textit{CalculationInterestingScore}}(L, U, k)$
$IL = \textbf{\textit{Ranking}}(L, ISF^L, ISP^L)$
$L^{nu} = \textbf{\textit{TopNGeneration}}(loc(nu), d, N, IL, L)$

⋆⋆⋆ ***Procedure CalculationInterestingScore***(L, U, DB, t)
• $ISF^U = \{isf(u_k) | u_k \in U \wedge isf(u_k) = 1\}$
• $ISP^U = \{isp(u_k) | u_k \in U \wedge isp(u_k) = 1\}$
for all i^{th} iteration where $i \leq t$ **do**
• $ISF^L = \{isf(l_j) | l_j \in L \wedge isf(l_j) = \sum_{x=1}^{p}(f_x \times isf(u_x))\}$ (eq. 1)
• $ISP^L = \{isp(l_j) | l_j \in L \wedge isp(l_j) = \sum_{x=1}^{p}(f_x \times pref(u_x) \times isp(u_x))\}$ (eq. 7)
• $ISF^U = \{isf(u_k) | u_k \in U \wedge isf(u_k) = \sum_{y=1}^{p}(f_y \times isf(l_y))\}$ (eq. 2)
• $ISP^U = \{isp(u_k) | u_k \in U \wedge isp(u_k) = \sum_{y=1}^{q}(f_y \times pref(l_y) \times isp(l_y))\}$ (eq. 9)
• normalize interesting scores in $ISF^L, ISF^U, ISP^L, ISP^U$ (eq. 3, 4, 10 and 11)
end for

⋆⋆⋆ ***Procedure Ranking***(L, ISF^L, ISP^L)
• $IL = \emptyset, rank = 1$
while $L \neq \emptyset$ **do**
• $max^{freq} = max(isf(l_1), isf(l_2), \ldots, isf(l_{|L|}))$
• $max^{pref} = max(isp(l_1), isp(l_2), \ldots, isp(l_{|L|}))$
• $L^{freq} = \{l_j \in L | isf(l_j) = max^{freq}\}$
• $L^{pref} = \{l_k \in L | isp(l_k) = max^{pref}\}$
if $max^{freq} = max^{pref}$
• $IL = IL \cup \{<l_j, rank> | l_j \in L^{freq}\}$
• $IL = IL \cup \{<l_k, rank> | l_k \in L^{pref}\}$
else if $max^{freq} > max^{pref}$
• $IL = IL \cup \{<l_j, rank> | l_j \in L^{freq}\}$
• $rank{+}{+}$
• $IL = IL \cup \{<l_k, rank> | l_k \in L^{pref}\}$
else
• $IL = IL \cup \{<l_k, rank> | l_k \in L^{pref}\}$
• $rank{+}{+}$
• $IL = IL \cup \{<l_j, rank> | l_j \in L^{freq}\}$
end if
• $ISF^L = ISF^L - \{isf(l_j) | l_j \in L^{freq}\}$
• $ISP^L = ISP^L - \{isf(l_k) | l_k \in L^{pref}\}$
• $L = L - (L^{freq} \cup L^{pref})$
• $rank{+}{+}$
end while

⋆⋆⋆ ***Procedure TopNGeneration***$(loc(nu), d, N, IL, L)$
• $L^{nu} = \emptyset, rank = 0$
for each location l_j in IL and $rank < N$ **do**
if $diff(loc(nu), loc(l_j)) \leq z$
• $L^{nu} = L^{nu} \cup \{l_k | l_k \in IL, rank_{l_k} = rank_{l_j}\}$
• $rank{+}{+}$
end if
end for

from our recommendation system. The results are then compared with those of the HITS-based model. Lastly, we provide a performance analysis and related discussion.

4.1 Real-world datasets

For our experiments, we use a real-world dataset provided by Foursquare. The dataset [12] contains users' location-check-in histories. Each data entry consists of user ID, location ID, location category ID, a location category name, latitude, longitude, time zone offset, and UTC time collected from 12 April 2012 to 16 February 2013 (covering approximately 10 months). We focus only on locations in the most popular category (food category). Via Google Maps Geocoding API[1], we transform longitude and latitude data into the names of the physical areas. We then select five areas (including Chiyoda, Shibuya, Minato, Shinjuku and Chuo) in Tokyo and the New York area. These areas' associated data contain the highest frequency check-ins (i.e., the largest number of location check-ins) for this study. Since this dataset size is large and very sparse, we pre-process the dataset by discarding users who visited fewer than two locations, and locations each of which had fewer than two visitors. The remaining dataset is divided into a training dataset (80% of the total number of the remaining users) and a testing dataset (20%) for five-fold cross validation. In the testing phase, the users in the testing dataset are regarded as new users (to simulate cold-start users). Note that our method does not require using a new users' profile to determine potential locations. The characteristics of the dataset are shown in Table 2.

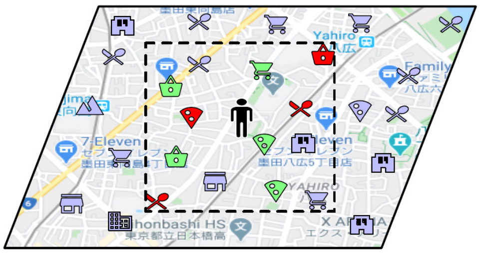

Fig.4: The example of current location simulation of a user.

4.2 Experiment settings and evaluation method

We compare our method with the HITS-based model (as adopted by [1] and [2]). Other than being classic, the HITS-based model shares the same target as ours. It aims to recommends locations to new users (who have no profile or check-in histories). The Chiyoda, Shinjuku, Minato, Shinjuku, Chuo and New York datasets are sparse, so we define that the

[1] https://developers.google.com/maps/documentation/geocoding/

Table 2: The characteristics of the dataset

City	Users	Locations	Total check-ins	Check-ins per location	Check-ins per user
Chiyoda	958	827	7,283	8.81	7.60
Shibuya	581	719	4,107	5.71	7.10
Minato	519	719	3,836	5.34	7.39
Shinjuku	571	670	3,905	6.84	5.83
Chuo	243	325	1,645	5.06	6.77
New York	921	3,609	26,494	7.34	28.77

largest number of recommended locations as 10 (Top-N, where N is 10). From this, we consider recommending locations only to the users who have more than 10 candidate locations. In all experiments, a location in the testing dataset can be suggested only if at least one user had visited or checked in the location more than 1 time. As users' current locations are not specified in the dataset, we simulate them by creating each user's minimum bounding rectangle (MBR, shown as the dashed line in Fig. 4) ([1] and [13]). All locations located in a MBR are considered to determine whether or not they should be recommended to the user. In Fig. 4, we show an example of the current-location simulation using MBR in which the ground-truth locations are shown in red, and the recommended ones are shown in green. The algorithm performs perfectly only if it can recommend all the ground-truth locations (which are the ones the user visited at least once). The data related to ground-truth locations is not included in the training process.

For measuring the accuracy, we use four metrics reflecting the correctness of the recommended top-N locations. Specifically, we use precision, recall, average ranking, and NDCG, as shown in Equations 12, 13, 14, and 15 respectively.

$$Precision = \frac{\#correct\_retrieved}{N} \tag{12}$$

$$Recall = \frac{\#correct\_retrieved}{\#relevant} \tag{13}$$

$$AverageRanking = \frac{\sum rank}{\#testloc} \tag{14}$$

$$NDCG = \frac{DCG}{IDCG} \tag{15}$$

N is the number of recommendations. $\#correct\_retrieved$ is the number of ground-truth locations retrieved correctly. $\#relevant$ is the number of all ground-truth locations. $\#testloc$ is the number of the ground-truth locations recommended by NILR or the HITS-based model. $rank$ is the position of the ground-truth locations in the recommendation list. DCG denotes the discounted cumulative gain (i.e., the position score of the ground-truth locations in the recommended list, [15] $IDCG$ denotes the idealized discounted cumulative gain (i.e., the perfect position score of the locations in the ideal recommendation list, [15])

4.3 Results and Discussion

Examining the experimental results, we analyze the accuracies of the recommendation lists in terms of precision, recall, average ranking, and NDCG against the varied number of recommendations. As shown in Figures 5 and 6, we compare the accuracies of our recommendation lists with those of the HITS-based model in terms of precision and recall, respectively. The performance comparisons measured by both metrics are similar. The precision and recall values obtained by our method are higher on average in all areas (except for the Shinjuku area). That anomaly occurs is because users in Shinjuku usually go to the shops of high check-in frequencies but have a low number of revisits, complying with the HITS-based model. In contrast, users in the other areas also users frequently revisited shops. Consequently, in the latter cases, our method outperforms the HITS-based model. Furthermore, we observe that the precision and recall values are quite low in our NILR approach and also the traditional HITS-based model as a result of the fact that we designed the offline experiments based on the check-in histories provided by Foursquare [12]. The evaluations were done by comparing the recommendation lists against the check-in histories (instead of being suggested to actual users). If we were to evaluate our method with real users, we would expect higher accuracy as the users would be able to look up and would be inclined to choose the locations suggested by the lists. In another case, the Foursquare users might actually have gone to certain locations being recommended by NILR but the subjects have forgotten to check-in via Foursquare. Those locations were excluded from the set of ground-truth locations in our experiments, resulting in the low precision and recall values. In summary, compared with the HITS-based model, our method yields the precision /recall values 6.37%/6.13%, and 24.43%/30.20% higher in Tokyo and New York, respectively. In terms of ranking ac-

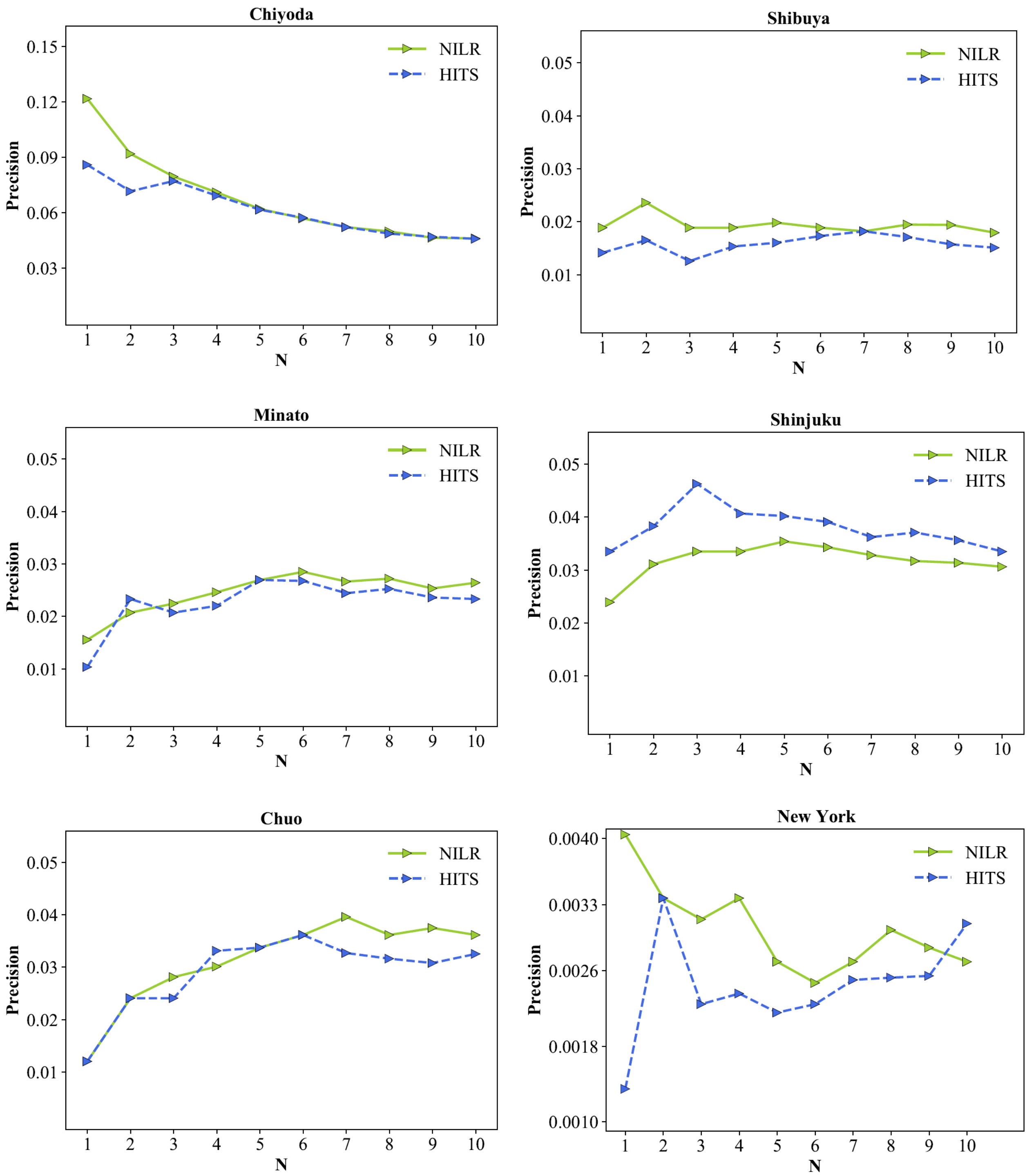

Fig.5: Precision metric w.r.t. Recommendation numbers.

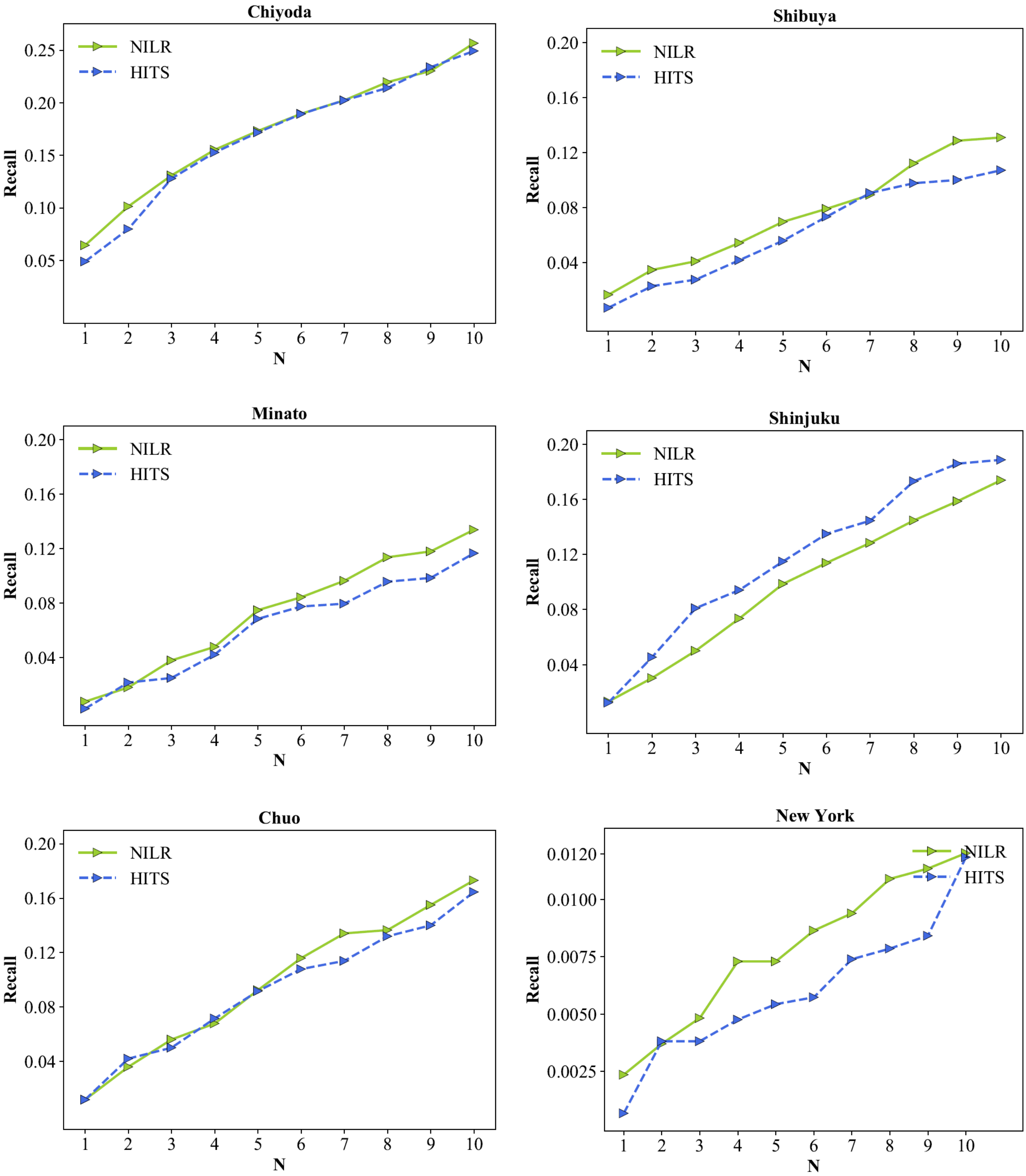

Fig.6: Recall metric w.r.t. Recommendation numbers.

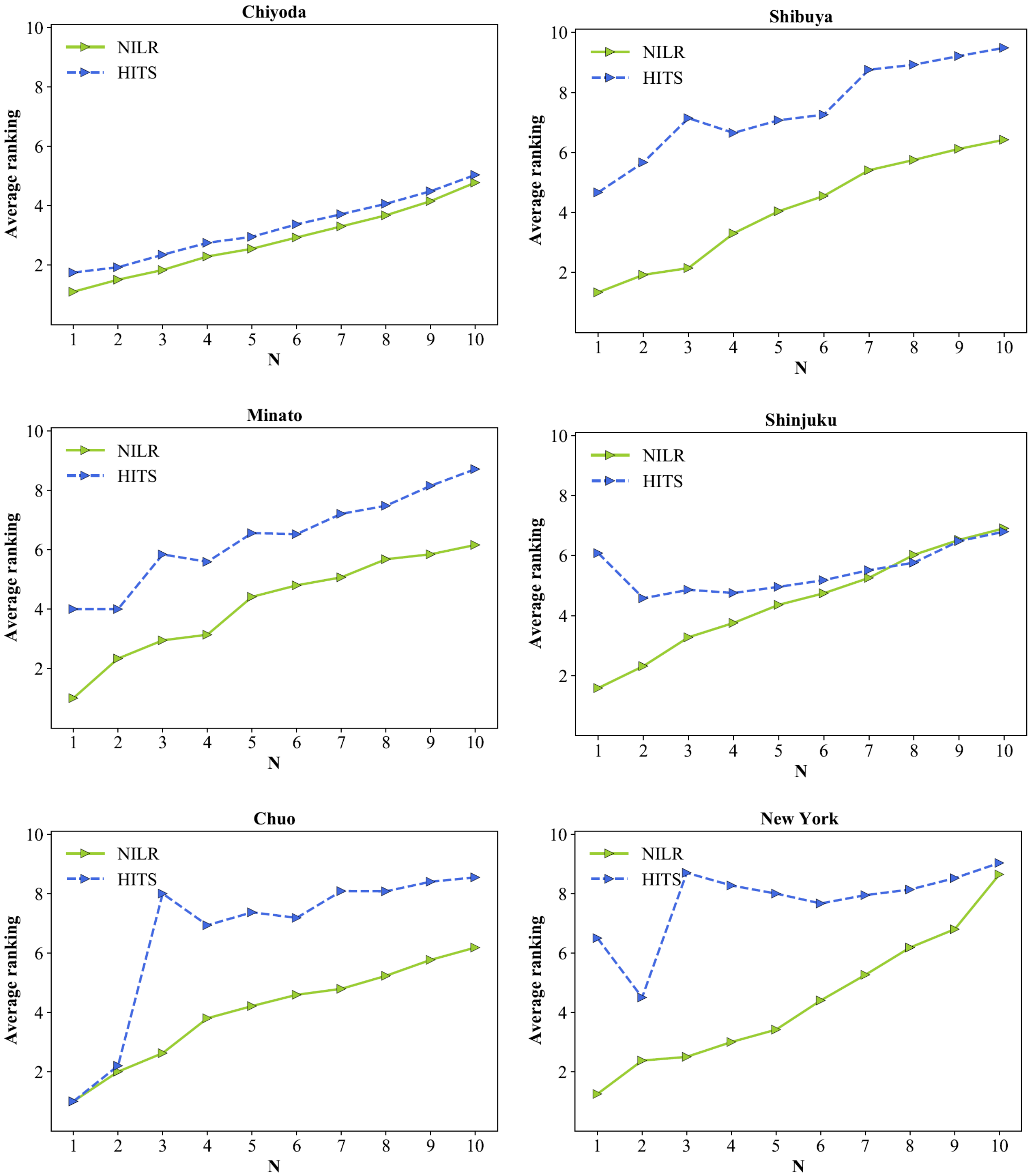

Fig. 7: Average ranking metric w.r.t. Recommendation numbers.

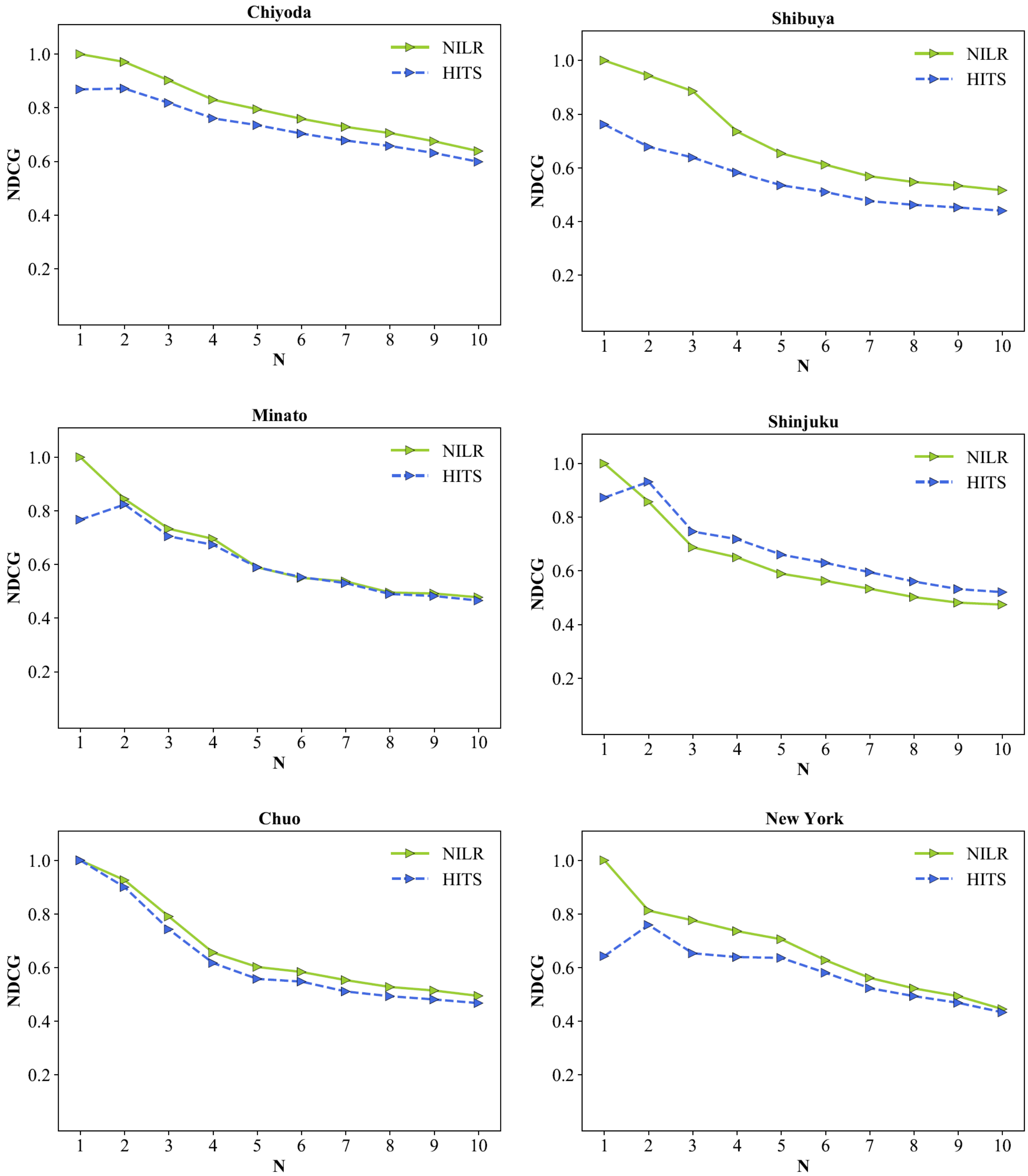

Fig.8: NDCG metric w.r.t. Recommendation numbers

curacy, we show the values of average ranking of the recommendation lists in Fig. 7. We can see that our method ranks the ground-truth locations in all areas lower than those of the HITS-based model. The average ranking accuracy of our method is 30.27% and 43.28% more in Tokyo and New York, respectively. Finally, Fig. 8 shows the average ranking accuracy in terms of NDCG. Our method outperforms the HITS-based model by 8.01% and 14.60% in Tokyo and New York, respectively.

The comparison results measured in terms of precision, recall, and NDCG are similar. The HITS-based model outperforms ours (by giving lower ranks to the ground-truth locations) in the Shinjuku area. However, if we use the average ranking as the metric, the comparison results are different, especially in that area. This is due to the fact that calculating the average ranking is based on the average of non-normalized ranks, while determining the NDCG is based on ranking numbers normalized by log scale. In most of the cases in the Shinjuku area, the HITS-based model outperforms our method by just a little. However, there are a few cases in the area where our method outperforms the HITS-based model by a large margin. NDCG normalizes the differences with a log scale which makes the large gaps smaller. As a consequence, the HITS-based model outperforms ours on average in the Shinjuku area.

For the other areas, our method yields more accurate recommendation lists with better rankings. This is due to the fact that users often choose to visit locations revisited many times (defined as *locations based on preference* in Section 3) as opposed to the places visited many times by various people (defined as *locations based on frequency* in Section 3). Our method retrieves and recommends locations which are outstanding in both the frequency and preference aspects while the HITS-based model is good at retrieving locations outstanding only in the frequency aspect. In some cases where both methods can retrieve and recommend the same locations, when considering the preference aspect, our method assigns lower ranks to the ground truth locations.

5. CONCLUSION

In this work, we introduce an enhanced location-recommendation approach called N-most interesting location-based recommender system (NILR) to recommend interesting locations for new users to visit. Our method includes three phases: 1) interesting locations are extracted by regarding both visiting frequencies and users' preference, 2) interesting locations are ranked based on two interestingness scores: one obtained from the HITS-based model and the other from our proposed method, and 3) interesting locations are re-filtered based on the current location of each new user. Experimental results from a well-known real-world dataset prove our NILR method outperforms the HITS-based model in terms of recommendation accuracy and ranking. Our method yields more accurate rankings than that of the HITS-based model by 30.27% and 43.28% for ranking recommended places in Tokyo city and New York respectively.

ACKNOWLEDGEMENT

This work was supported by Faculty of Informatics, Burapha University.

References

[1] J. Bao, Y. Zheng and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *Proceedings of the 20th international conference on advances in geographic information systems*, pp. 199-208, 2012.

[2] H. Bagci and P. Karagoz, "Context-aware location recommendation by using a random walk-based approach," *Knowledge and Information Systems*, vol.47, no.2, pp.241-260, 2016.

[3] Y. Zheng and X. Zhou, *Location-based social networks: Users*, pp. 243-276, 2011.

[4] Y. Zheng, L. Capra, O. Wolfson and H. Yang. "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol.5, no.3, Sep 2014.

[5] Y. Yu and X. Chen, "A survey of point-of-interest recommendation in location-based social networks," in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[6] S. Xu, X. Fu, J. Cao, B. Liu and Z. Wang, "Survey on user location prediction based on geo-social networking data," *World Wide Web*, pp. 1-44, 2020.

[7] D. Zhou, B. Wang, S. M. Rahimi and X. Wang, "A study of recommending locations on location-based social network by collaborative filtering," In Leila Kosseim and Diana Inkpen, editors, Advances in Artificial Intelligence, pages 255266, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[8] J. J. Levandoski, M. Sarwat, A. Eldawy and M. F. Mokbel, "Lars: A location-aware recommender system," in *2012 IEEE 28th International Conference on Data Engineering*, pp. 450-461, 2012.

[9] Zhijun Ding, Xiaolun Li, Changjun Jiang, and Mengchu Zhou. Objectives and state-of-the-art of location-based social network recommender systems. ACM Comput. Surv., 51(1), January 2018.

[10] Y. Zheng, L. Zhang, X. Xie and W.Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in*Proceedings of the 18th international conference on World wide web*, pp. 791-800, 2009.

[11] S. Darapisut, K. Amphawan, S. Rimcharoen and N. Leelathakul, "Nilr: N-most interesting

location-based recommender system," in *Conference on Smart Media and Applications*, 2020.

[12] D. Yang, D. Zhang, V. W. Zheng and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol.45, no.1, pp.129-142, 2014.

[13] Y. Ying, L. Chen and Gencai Chen, "A temporal-aware poi recommendation system using context-aware tensor decomposition and weighted hits," *Neurocomputing*, vol.242, pp.195-205, 2017.

[14] R. Burke, *Hybrid Web Recommender Systems*, pp. 377-408, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[15] C. Biancalana, F. Gasparetti, A. Micarelli and G. Sansonetti, "An approach to social recommendation for context-aware mobile services," *ACM Trans. Intell. Syst.Technol.*, vol.4, no.1, pp.1-31, Feb. 2013.

[16] S. Zhao, I. King and M. R. Lyu, "A survey of point-of-interest recommendation in location-based social networks," *arXiv preprint*, arXiv:1607.00647, 2016.

[17] X. Xiao, Y. Zheng, Q. Luo and X. Xie, "Finding similar users using category-based location history," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 442-445, New York, USA, 2010.

[18] P. Symeonidis, A. Papadimitriou, Y. Manolopoulos, P. Senkul, and I. Toroslu, "Geo-social recommendations based on incremental tensor reduction and local path traversal," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pp. 89-96, New York, USA, 2011.

[19] H. Wang, G. Li and J. Feng, "Group-based personalized location recommendation on social networks," in *Lei Chen, Yan Jia, Timos Sellis, and Guanfeng Liu, editors, Web Technologies and Applications*, pp. 68-80, Cham, 2014. Springer International Publishing.

[20] Q. Yuan, G. Cong, Z. Ma, A. Sun and N. Ma. Thalmann, "Time-aware point-of-interest recommendation," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13*, pp. 363-372, New York, USA, 2013.

[21] K. H. Lim, J. Chan, C. Leckie and S. Karunasekera, "Personalized trip recommendation for tourists based on user interests," *points of interest visit durations and visit recency*, vol.54, no.2, pp. 375-406, February 2018.

[22] K. H. Lim, J. Chan, S. Karunasekera and C. Leckie, "Tour recommendation and trip planning using location-based social media: a survey," *Knowledge and Information Systems*, pp. 1-29, 2019.

[23] V. W. Zheng, B. Cao, Y. Zheng, X. Xie and Q. Yang, "Collaborative filtering meets mobile recommendation: A user-centered approach," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI'10*, pages 236-241. AAAI Press, 2010.

[24] V. W. Zheng, Y. Zheng, X. Xie and Q. Yang, "Collaborative location and activity recommendations with gps history data," in *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pp. 1029-1038, New York, USA, 2010.

[25] X. Liu, Y. Liu, K. Aberer and C. Miao, "Personalized point-of-interest recommendation by mining users' preference transition," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13*, pp. 733-738, New York, USA, 2013.

[26] D. Preoţiuc-Pietro and T. Cohn, "Mining user behaviours: a study of check-in patterns in location based social networks," in *Proceedings of the 5th annual ACM web science conference*, pp. 306-315, 2013.

[27] Y. Si, F. Zhang and W. Liu, "Ctfara: An adaptive method for poi recommendation based on check-in and temporal features," *Knowledge-Based Systems*, vol.128, pp.59-70, 2017.

[28] J. Cao, S. Xu, X. Zhu, R. Lv and B. Liu, "Effective fine-grained location prediction based on user check-in pattern in lbsns," *Journal of Network and Computer Applications*, vol.108, pp.64-75, 2018.

[29] R. Baral and T. Li, "Exploiting the roles of aspects in personalized poi recommender systems," *Data Mining and Knowledge Discovery*, vol.32, no.2, pp.320-343, 2018.

[30] M. Gan and L. Gao, "Discovering memory-based preferences for poi recommendation in location-based social networks," *ISPRS International Journal of Geo-Information*, vol.8, no.6, 279, 2019.

[31] M. Xin, L. Wu and S. Li, "A user profile awareness service collaborative recommendation algorithm under lbsn environment," *International Journal of Cooperative Information Systems*, 28(03):1950008, 2019.

[32] J. M. Kleinber,. "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol.46, no.5, pp.604-632, 1999.

[33] X. Long and J. Joshi. "A hits-based poi recommendation algorithm for location-based social networks," in *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pp. 642-647, 2013.

[34] J. Chen, X. Li, W.K. Cheung and K. Li. A survey of point-of-interest recommendation in location-

based social networksEffective successive poi recommendation inferred with individual behavior and group preference," *Neurocomput.*, 210 pp.174-184, Oct. 2016.

[35] D. Puspitaningrum, J. Fernando, E. Afriando, F. P. Utama, R. Rahmadini and Y. Pinata, "Finding Local Experts for Dynamic Recommendations Using Lazy Random Walk," *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1-6, 2019.

Sumet Darapisut received BS and MS degrees in Computer Science in 2013 and 2016, respectively. Currently, he is working toward the PhD degree in the Faculty of Informatics, Burapha University, Thailand. His research interests include recommender system and data mining.

Komate Amphawan received BS and MS degrees in Computer Science in 2003 and 2006, respectively. Then, he received Ph.D. degree in Computer Engineering in 2011. Since 2011, he has been a faculty member in Faculty of Informatics, Burapha University, Thailand. His research interests include the area of data mining and knowledge discovery.

Sunisa Rimcharoen received a B.Sc. degree in computer science from Burapha University, Thailand, in 2003, an M.Sc. degree in computer science from Chulalongkorn University in 2005, and a Ph.D. degree in computer engineering from Chulalongkorn University in 2009. Dr. Sunisa has been a faculty member of the Faculty of Informatics, Burapha University, Chon Buri, Thailand, since 2009. Her current research interests include evolutionary computation, machine learning, prediction, and algorithmic trading.

Nutthanon Leelathakul received a B.Eng. degree with first-class honors in electrical engineering from Kasetsart University, Bangkok, Thailand, in 1999, an M.Eng. degree in electrical and computer engineering from Cornell University in 2004, and a Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 2010. Dr. Nutthanon has been a faculty member of the Faculty of Informatics, Burapha University, Chon Buri, Thailand, since 2010. His current research interests include computer networking, security in IoT, deep learning, and data analytics.