

Mining User Check-in Behavior with a Random Walk for Urban Point-of-interest Recommendations

JOSH JIA-CHING YING, National Cheng Kung University

ERIC HSUEH-CHAN LU, National Taitung University

WEN-NING KUO, National Cheng Kung University

VINCENT S. TSENG, National Cheng Kung University

In recent years, research into the mining of user check-in behavior for point-of-interest (POI) recommendations has attracted a lot of attention. Existing studies on this topic mainly treat such recommendations in a traditional manner, i.e., they treat POIs as items and check-ins as ratings. However, users usually visit a place for reasons other than to simply say that they have visited. In this paper, we propose an approach called Urban POI-Walk (UPOI-Walk), which takes into account a user's Social-triggered Intentions, Preference-triggered Intentions, and Popularity-triggered Intentions, to estimate the probability of a user checking-in to a POI. The core idea of UPOI-Walk involves building a HITS-based random walk on the normalized check-in network, thus supporting the prediction of POI properties related to each user's preferences. To achieve this goal, we define several user-POI graphs to capture the key properties of the check-in behavior motivated by user-intentions. In our UPOI-Walk approach, we propose a new kind of random walk model named Dynamic HITS-based Random Walk, which comprehensively considers the relevance between POIs and users from different aspects. On the basis of similitude, we make an online recommendation as to the POI the user intends to visit. To the best of our knowledge, this is the first work on urban POI recommendations that considers user check-in behavior motivated by Social-triggered Intentions, Preference-triggered Intentions, and Popularity-triggered Intentions in location-based social network data. Through comprehensive experimental evaluations on two real datasets, the proposed UPOI-Walk is shown to deliver excellent performance.

Categories and Subject Descriptors: **H.2.8 [Database Management]**: Database Applications—*Data Mining*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Point-Of-Interest Recommendation, Urban Computing, Data Mining, Location-Based Social Network, User Preference Mining.

ACM Reference Format:

Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wen-Ning Kuo and Vincent S. Tseng, YYYY. Mining User Check-in Behavior with a Random Walk for Urban Point-of-interest Recommendations. *ACM Trans. Intelligent Systems and Technology*. xx, x, Article xx (Month YYYY), P pages.

DOI:

This research was supported by National Science Council, Taiwan, R.O.C. under grant no. NSC101-2221-E-006-255-MY3. This article is an expanded version of [Ying *et al.*, 2012], which appeared in Proceedings of the ACM International Workshop on Urban Computing.

Author Addresses: Josh Jia-Ching Ying, Wen-Ning Kuo, and Vincent S. Tseng, Institute of Computer Science and Information Engineering, National Cheng Kung University, No.1, University Road, Tainan City 701, Taiwan (R.O.C.); email: tsengsm@mail.ncku.edu.tw.

Eric Hsueh-Chan Lu, Department of Computer Science and Information Engineering, National Taitung University, No.684, Sec. 1, Zhonghua Rd., Taitung City, Taitung County 95002, Taiwan (R.O.C.); email: luhc@nttu.edu.tw.

Permission to make digital or hardcopies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credits permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2013 ACM 2157-6904/YYYY/xx-ARTxx \$15.00

DOI:

1. INTRODUCTION

Many urban applications, such as navigational planning services [Yuan *et al.*, 2011], urban planning [Yuan *et al.*, 2012], and location-based advertisement [Lu *et al.*, 2012], have been developed for the rapidly growing location-based services (LBSs) market. To improve the quality of smart urban living, it is beneficial that these LBSs can recommend to their users ‘Points-Of-Interest’ (POIs) in which they may be interested. Thus, effective and efficient urban POI recommendation techniques are desirable. Given a set of POIs, which is pre-determined via a location-based social network (LBSN) [Zheng *et al.*, 2012] [Zheng and Zhou, 2011], the recommendation technique identifies the list of POIs a user is most likely to visit. Intuitively, to model the relevance between the users and the POIs, we can construct a user-POI matrix in which rows denote the users and columns denote the POIs. Each entry in this user-POI matrix denotes a relevance score, showing the probability that the user will visit the POI. Therefore, as shown in Fig. 1, we can see POI recommendations given some user query as a ranking over the rows of the matrix. Intuitively, each entry in the user-POI matrix could be estimated by users’ check-ins. However, users typically only check-in to a small portion of all POIs. Consequently, we need to calculate a predicted score for the POIs that a user has not visited.

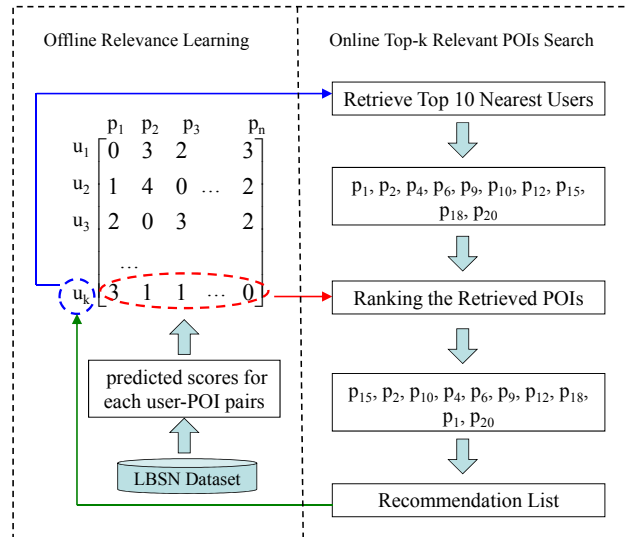


Fig. 1. An example of POI recommendation.

To address the calculation of these predicted scores for the user-POI pairs, Random Walk approaches have shown an excellent performance [Jamali and Ester, 2009; Cao *et al.*, 2010]. Using the Random Walk approach to predict scores for the user-POI pairs, a weighted graph is required. Generally, the weights associated with each edge of the graph should reflect the intention of the user to visit the POI [Zheng, 2012]. On the basis of our observations, we categorize users’ intentions into three classes:

- *Social-triggered Intentions (SI)*, which refer to the relationship between POIs and a user’s social circle. These intentions reflect the frequency with which the user’s social circle visits the POI. For example, as shown in Fig. 2, we could recommend a Chinese restaurant to user A, because his/her friends often check-in to such restaurants.

- *Preference-triggered Intentions (PreI)*, which refer to the relationship between the semantic labels of the POI and a user's preferences. These intentions reflect how often the user visits POIs with a specific semantic label. As shown in Fig. 2, user B tends to check-in to POIs in the "coffee" category. Therefore, we may recommend other POIs in this category.
- *Popularity-triggered Intentions (PopI)*, which refer to the relationship between POIs and their number of check-ins. These intentions reflect the general phenomenon whereby people like to visit popular places. For example, as shown in Fig. 2, most users go to Chinese restaurants for a meal. We thus recommend Chinese restaurants to other users.

Corresponding to these three types of intention, users' check-ins can be viewed as the context of the behavior that is motivated by the aforementioned intentions. In other words, check-ins may be triggered by multiple intentions, inclusive of SI, PreI, and PopI. However, most existing techniques [Ye *et al.*, 2010; Ye *et al.*, 2011b] for recommending POIs are primarily focused on analyzing SI for enhancing collaborative filtering (CF)-based models. To extract the check-in behavior motivated by SI, existing methods usually extract social-based features from LBSN data. Due to the social binding of a user's social circle, these recommendations tend to recommend the POIs that usually have been visited by the user's friends. For example, as shown in Fig. 2, users A and B have both visited coffee shops, implying that their preferences in restaurants are very similar. Suppose that user B likes to eat pizza. Thus, it is more reasonable to recommend a pizza parlor than a Chinese restaurant to user A. But the social inference recommendation models would recommend Chinese restaurant to user A. The reason is that these existing models do not address users' preferences from semantic tags of visited POIs. Hence, we exploit users' preferences from semantic tags of visited POIs to recommend the POIs of LBSNs in an urban area.

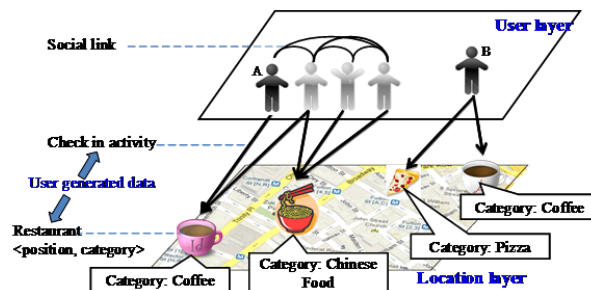


Fig. 2. Location-based social network.

Several works have addressed the check-in behavior motivated by PreI [Bao *et al.*, 2012; Cao *et al.*, 2010]. As mentioned earlier, semantic label plays a crucial role for representing users' preference intention in urban POI recommendation. Unfortunately, these works only consider the pre-defined category of the POI as a rough semantic label while disregarding the user-generated annotation of the POI. Although we can also map users' current positions to some POI database or road network in order to understand the POI tags given by users, this process does not work well in an urban area. This is because POIs in an urban area are very crowded, which means we cannot accurately tag a location just by determining users' positions. Fig. 3 shows that there are two totally different types of restaurant in the same building. Fortunately, we can view the LBSN of each user as a sensor for detecting

semantic POI tags, thus improving POI recommendation in an urban area. Moreover, the check-in behavior motivated by PopI has still not been taken into consideration in the representation of user behavior. As mentioned earlier, users generally visit only a small portion of all POIs. This leads to the features extracted from the SI and PreI behavior being ineffective for recommending POIs that the user and his/her social circle never visit.



Fig. 3. A POI scenario in an urban area.

On the basis of the observations prompted by the above examples, we propose *Urban POI-Walk (UPOI-Walk)*. This is a new approach to urban POI recommendation that mines urban users' check-in behavior based on three kinds of intentions. As shown in Equation (1), given a set of users U and a set of POIs P , the problem of POI recommendation can be formulated as the prediction of relevance scores, i.e., the degree of interest, for a given user-POI pair.

$$f(u, p) \rightarrow v, \text{ where } u \in U, r \in P, \text{ and } v \in [0, 1] \quad (1)$$

Hence, POI recommendation in LBSNs can be addressed as the problem of numerical value prediction. Although numerical value prediction techniques have been developed for many applications, such as demographic prediction [Hu *et al.*, 2007], bio life-cycle analysis [Menzies *et al.*, 2003], and the prediction of natural phenomena [Etemad-Shahidi and Mahjoobi, 2009], they have not yet been explored in the context of urban computing. In addition, the question of extracting significant features to support POI recommendations based on heterogeneous check-in intentions is also a critical and challenging issue. In building a numerical value prediction model, the fundamental issue is handling the heterogeneous features extracted from the heterogeneous data. As mentioned earlier, Random Walk approaches display an excellent performance in calculating predicted scores for the user-POI pairs. However, existing works [Jamali and Ester, 2009; Cao *et al.*, 2010] do not address such heterogeneous features, preferring instead a straightforward aggregation of the extracted features, such as by linear combination. This leads to some features dominating over others.

To address the above-mentioned problem, we explore the movements of users triggered by *SI*, *PreI*, and *PopI*, and seek several descriptive features to explain the relevance of user-POI pairs. In contrast to conventional POI recommendation techniques, which are based solely on the check-in behavior motivated by *SI*, we also utilize the check-in behavior motivated by *PreI* and *PopI* to recommend POIs for LBSN users. Our *Urban POI-Walk* method follows the HITS-based Random Walk approach [Cao *et al.*, 2010]. This comprises: 1) an offline relevance learning module, and 2) an online top- k relevant POIs search. To learn the relevance of each user-POI pair, the relevance learning module explores all social, personal preference, and popularity aspects of the POI check-ins being captured in the LBSN dataset. The

relevance learning module includes three main steps: (i) feature extraction—the extraction of significant features of user-POI pairs in the forms of SI, PreI, and PopI; (ii) user-POI graph construction—the construction of weighted bi-partite graphs based on the extracted features. As the main idea underlying the HITS-based Random Walk approach is to predict the relevance of a user-POI pair in terms of the user’s ability to reach the POI, the user-POI graph plays a crucial role in the random walk model. Finally, (iii) Dynamic HITS-based Random Walk—calculation of the relevance of each user-POI pair by performing a HITS-based Random Walk with dynamic selection of the constructed user-POI graph. In the online recommendation module, the relevance between the properties of the POI and the user’s preference can be used to rank the recommendation list of POIs. After checking their coordinates, we obtain all qualified POIs for a particular user.

This research makes a number of significant contributions, which are summarized as follows:

- We formulate the problem of POI recommendation in an urban area as the problem of relevance score prediction. This problem has not previously been explored in the research community.
- We propose *Urban POI-Walk (UPOI-Walk)*, a new approach for urban POI recommendation, by mining urban users’ check-in behavior. The proposed UPOI-Walk learns a random walk to estimate the relevance score of each user-POI pair. UPOI-Walk simultaneously explores three intentions of user check-in behavior, namely i) *Social-triggered Intentions*, ii) *Preference-triggered Intentions*, and iii) *Popularity-triggered Intentions*, by exploiting LBSN data to extract descriptive features.
- We propose an extension to the HITS-based Random Walk model, called *Dynamic HITS-based Random Walk*, for estimating the relevance between POIs and users. The proposed Dynamic HITS-based Random Walk model can deal with the heterogeneous feature problem well by building several weighted complete bi-partite graphs.
- We use two real datasets to evaluate the performance of UPOI-Walk in a series of experiments. One is obtained from Gowalla (<http://www.Gowalla.com/>), and the other is from EveryTrail (<http://www.EveryTrail.com/>). The results show that UPOI-Walk delivers superior performance over other recommendation techniques in terms of the popular normalized discounted cumulative gain and mean absolute error measures.

The rest of this paper is organized as follows. We briefly review the related work in Section 2 and provide overview our urban POI recommendation approach UPOI-Walk in Section 3. In section 4, the details of our proposed features are described. The proposed Dynamic HITS-Based Random Walk model is detailed in section 5. Finally, we present the evaluation result of our empirical performance study in Section 6 and discuss our conclusions and future work in Section 7.

2. RELETED WORK

In this chapter, we review relevant previous studies and classify them into four categories: i) General Item Recommendation Systems, ii) Location or POI Recommendation, iii) Spatial-Temporal-based Data Mining, and iv) Numerical Value Prediction.

2.1 General Item Recommendation Systems

2.1.1 Collaborative Filtering (CF)-based models. To recommend general items to users, Collaborative Filtering (CF)-based methods are highly effective. Generally, CF-based recommendations recommend items for users by using the records of similar users or similar items. Horozov et al. [Horozov et al. 2006] proposed an enhanced CF solution for personalized POI recommendations. Some approaches [Massa and Avesani, 2007] have proposed a trust-aware recommendation system, which builds a trust metric and makes use of trust information to recommend items. Debnath et al. [Debnath et al., 2008] proposed a combined CF and content-based recommendation system. Their approach can determine the weight values of attributes, which are obtained from a social network, using linear regression. To solve the problem of sparse data, many studies use CF- and item-based approaches to estimate missing values. The fundamental problem of the CF-based approach is the evaluation of user similarity and location similarity. Spertus et al. [Spertus et al., 2005] proposed six different measures for recommending online social networks. The six measures applied the cosine distance, mutual information measure, TF-IDF, and log-odds functions to measure the similarity of communities. In [Ye et al., 2012b], Ye et al. proposed probabilistic generative model which naturally unifies the ideas of social influence, collaborative filtering and content-based methods for item recommendation.

2.1.2 Social-based models. General Speaking, social-based recommendations consider that a user's interests would be influenced by his/her friends. Accordingly, social-based recommendations recommend items for a user by using the records of the user's social circle. Jamali and Ester [Jamali and Ester, 2009] proposed a new kind of random walk-based model, called TrustWalker. The TrustWalker model performs the random walk on the social network to predict item ratings for users. Trust-based random walk uses trust values and user-to-item ratings to predict the item ratings. A trust value is the user-to-user rating. If two people's interests are similar, they will have a high mutual trust value. TrustWalker can solve the "cold start problem" with the same precision.

2.1.3 Content- or Geo-aware Recommender. Levandoski et al. [Levandoski et al., 2012] considered the user location and item location to be a constraint. They believed that users' preferences were related to their location, and that users usually move short distances. Therefore, considering user and item locations should improve the performance of item recommendation. Scellato et al. [Scellato et al., 2011] proposed a social link prediction model, which involves several geo-based features intended to improve the accuracy of prediction. Content-based recommendation systems [Ono et al., 2007] use Bayesian network modeling of user preferences to recommend, e.g., movies. Bayesian networks are highly flexible, making them appropriate for representing the complex relations between users' preferences and contexts.

2.2 Location or POI Recommendation

2.2.1 Location Recommendation. Similar to POI recommendation, a location recommendation system recommends a specific geographical place to users. The major difference between POI recommendation and location recommendation is that the notion of a location usually refers to a region that is detected by some heuristic, such as a holiday destination [Zheng et al., 2009]. In recent years, many studies have addressed the issue of location recommendation. Zheng et al. [Zheng et al., 2010] employed an activity term, such as hiking, as a query term to request the location recommendation service. A mobile application framework was presented by Simon

and Fröhlich [Fröhlich, 2007] to enable a mobile phone user to query geo-coded Wikipedia articles for landmarks in the vicinity, and a Cyberguide system [Abowd *et al.*, 1997] was developed to provide information describing nearby buildings and related people's identities. Takeuchi and Sugimoto [Takeuchi and Sugimoto, 2006] developed CityVoyager to recommend shops. This system collected users' shop visiting histories based on GPS logs, and applied an item-based collaborative filtering method to recommend other shops similar to those previously visited. In order to recommending restaurants, a system combining both users' preferences and location contexts has also been studied in [Park *et al.*, 2007]. This used Bayesian learning to calculate recommendation values and provide a ranking list for restaurants. Zheng *et al.* [Zheng *et al.*, 2009] selected tourist hotspots as the recommended locations. A HITS-based model was proposed to account for users' travel experiences and locations of interest in forming the recommendations, so that only especially popular places that were also recommended by experienced users were ranked. Cao *et al.* [Cao *et al.*, 2010] utilized a HITS-based Random Walk to learn the significance of certain locations. They also used the Yellow Pages directories to understand the semantic labels of a location. Although such semantic labels are not annotated by users, they may at least partially represent the relationship between location and user preference. Leung *et al.* [Leung *et al.*, 2011] proposed a Collaborative Location Recommendation (CLR) framework based on co-clustering. This approach considers location, user, and activity to build a Community Location Model (CLM) graph. The CLR approach then uses Community-based Agglomerative-Divisive Clustering (CADC) algorithms to iteratively merge and divide nodes in the CLM graph. After clustering users, locations, and activities by CADC, refined clusters of similar locations that are visited by similar users doing similar activities are obtained.

2.2.2 POI Recommendation. Unlike location recommendation, POI recommendation is based on a pre-determined location. Ye *et al.* [Ye *et al.*, 2010] developed a friend-based CF approach for location recommendation based on collaborative POI ratings made by social friends. They also found that users' common check-ins were proportional to their geo-distance. Accordingly, Ye *et al.* proposed a geo-measured approach to approximate friend-based CF, thus reducing computation time. Later, Ye *et al.* [Ye *et al.*, 2011b] proposed a CF-based POI recommendation framework. This framework fused user preference influence, social influence, and geographic influence to infer the check-in probability for a given user at a POI. This approach also exploited a power-law distribution to build the geographical influence among POIs, and used the CF method to depict user preference influence and social influence. Bao *et al.* [Bao *et al.*, 2012] elicited social opinions from the location histories of local experts and learnt user personal preferences from users' location histories. Based on the social opinions and user personal preferences discovered, Bao *et al.* designed a location-based and preference-aware recommendation system that offers a particular user a set of venues within a geospatial range. Berjani and Strufe [Berjani and Strufe, 2011] proposed a Regularized Matrix Factorization (RMF) system to recommend appropriate spots, which could be viewed as POIs, for users. This research used data crawled from an LBSN website, Gowalla. The RMF recommender is a personalized application for recommending places. It first maps users and spots to a joint latent factor space, then exploits a regularized Singular Value Decomposition model to predict the ratings of users to spots.

2.3 Spatial-temporal-based Data Mining

To understand a user's preference, some research has been conducted that is devoted to mining location semantics or analyzing temporal information. For example, Ye *et al.* [Ye *et al.*, 2011a] proposed a semantic annotation approach in which user check-in behaviors and temporal information are utilized to form several useful features for improving the accuracy of semantic annotation. Ye *et al.* [Ye *et al.*, 2012a] conducted a series analysis of Temporal-Semantic-Interaction. Based on this analysis, several LBSN applications could be improved, such as Tag Recommendation, Place Selection, and Data Cleaning. Cho *et al.* [Cho *et al.*, 2011] analyzed LBSN and cellphone data to generate user mobility patterns. According to their analysis, humans experience a combination of strong, short-range spatially and temporally periodic movement that is not impacted by the social network structure, whereas long-distance travel is influenced more by social network ties. Such findings could be utilized to improve the accuracy of location prediction.

2.4 Numerical Value Prediction

Relevance estimation can be formulated as a numerical value prediction problem [Hu *et al.*, 2007; Menzies *et al.*, 2003; Etemad-Shahidi and Mahjoobi, 2009]. The typical steps in training a numerical value prediction model are: 1) feature extraction, 2) feature selection, and 3) model building. After selection, features are used as inputs for training a numerical value prediction model. Regression-tree models have shown an excellent performance in similar tasks [Etemad-Shahidi and Mahjoobi, 2009]. Once the model is trained, the training data is divided according to the decreasing variance of the target attribute. Each piece of training data is then used to build an individual regression model. Meanwhile, the procedure of dividing the training data is recorded to build a decision tree.

3. OVERVIEW OF URBAN POI-WALK

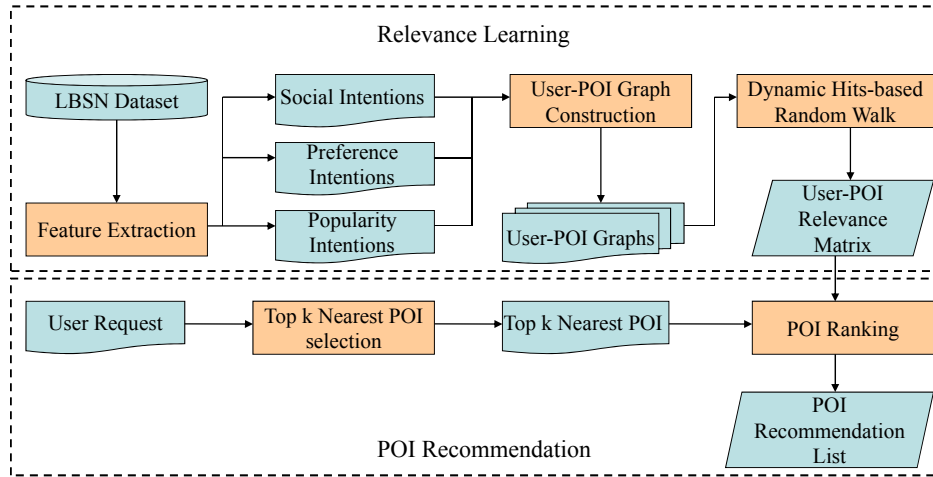


Fig. 4. UPOI-Walk approach.

To overcome the heterogeneous features and data sparsity problems, we design a two-stage approach, named Urban POI-Walk (UPOI-Walk), consisting of a relevance learning module and a POI recommendation module, as shown in Fig. 4. The system recommends urban POIs by mining user check-in behavior. In UPOI-Walk, the

relevance learning module deals with the relevance estimation of user-POI pairs, while the POI recommendation module handles the POI recommendations based on the estimated relevance of each user-POI pair. In the learning module, there are two main phases, i.e., feature extraction and relevance estimation.

For a specific user-POI pair, we explore the check-in behavior motivated by SI as population features that abstract the aggregated check-ins of the user's friends. We also explore the check-in behavior motivated by PreI between users and POIs to formulate the descriptive features of a given user-POI pair. Moreover, the check-in behavior motivated by PopI is considered for feature extraction in our recommendation model. The features derived from SI, PreI, and PopI are used to construct several complete bi-partite graphs. In the relevance estimation phase, we apply a Dynamic HITS-based Random Walk approach to these graphs to calculate the relevance score of each user-POI pair. We thus obtain a user-POI matrix that records the relevance between users and POIs. In the recommendation module, the relevance between users and POIs can be used to rank the POI recommendation list. After examining the coordinates of all POIs and user positions, we present all qualified POIs to the user.

4. RELEVANCE LEARNING

In the proposed UPOI-Walk approach, the relevance learning module is designed as a three-phase framework, as shown in Fig. 4. The first phase deals with feature extraction. This task explores the three types of intention discussed in the Introduction. The second phase handles the relevance estimation. We build two complete bi-partite graphs. Based on the extracted features, we can obtain two weighting functions to assign weights to the edges of the two graphs. Finally, in the third phase, we apply the Dynamic HITS-based Random Walk model to estimate the relevance between the user preferences and properties of the POI.

4.1 Feature Extraction

For a specific user-POI pair, we explore the check-in behavior motivated by SI, PreI, and PopI in order to explain the relevance between users and POIs.

4.1.1. Features from Social-triggered Intentions. Our goal is to extract discriminative social-triggered features from check-ins among the user's social circle at a POI. Intuitively, the user is likely to share some interests with his/her friends. Thus, if a POI is visited many times by the user's social circle, it may satisfy some of the user's interests. Accordingly, aggregating friends' relative check-ins at a POI can be used to infer the probability that a user likes that POI. Formally, given a friend f and a set of POIs P , f 's relative check-ins at POI p is formulated as

$$\text{relative check-ins}(f, p) = \frac{\text{checkins}(f, p)}{\sum_{p' \in P} \text{checkins}(f, p')} \quad (2)$$

However, it is possible there are users whose interests are totally different to those of his/her friends. Thus, the similarity between the user and his/her friends must be involved. Accordingly, given a user-POI pair (u, p) , the features extracted from SI can be generally formulated as

$$SI(u, p) = \sum_{f \in F(u)} (\text{relative check-ins}(f, p) \times \text{similarity}(f, u)) \quad (3)$$

where $F(u)$ is the set of user u 's friends.

As mentioned above, measuring the similarity of two users is the key feature obtained from SI. Intuitively, the spheres of users and their friends can be utilized to measure their similarity based on the nature of their lifestyles and activities offered by their living areas. As a result, different similarities, naturally formed by the aggregated behavior of friends with respect to various kinds of POI, are embedded in the friends' check-in activities. In LBSN data, the most important information concerns a user's common check-ins and the distance between users in terms of similarity. In the following, we extract two population features to depict POIs.

- Similarity by Common Check-ins (CheckSim): We employ the χ^2 test to ascertain the relation between the check-in behaviors of LBSN users and their friends. If the test shows that a relationship is significant, it means that the user typically visits a POI that her friends have also visited. The Gowalla dataset, shown in Fig. 5, exhibits this significant trait. Hence, common check-ins are good for measuring users' similarity. Hence, we formulate the similarity of two users as

$$\text{CheckSim}_{i,j} = \begin{cases} \cos(v_i, v_j) & , \text{ if user } i \text{ and } j \text{ are friends} \\ 0 & , \text{ otherwise} \end{cases} \quad (4)$$

where $\cos()$ indicates cosine similarity and v_i indicates the check-in vector of user i . Taking Table I as an example, the check-in vector of user i is $\langle 1, 0, 2, 5, 0 \rangle$, and the check-in vector of user j is $\langle 0, 10, 0, 1, 0 \rangle$. Thus, the $\text{CheckSim}_{i,j}$ is

$$\frac{(1 \times 0) + (0 \times 10) + (2 \times 0) + (5 \times 1) + (0 \times 0)}{\sqrt{1^2 + 0^2 + 2^2 + 5^2 + 0^2} \times \sqrt{0^2 + 10^2 + 0^2 + 1^2 + 0^2}} \approx 0.09$$

Table I. Example check-in log.

| POI ID | r1 | r2 | r3 | r4 | r5 |
|--------|----|----|----|----|----|
| User i | 1 | 0 | 2 | 5 | 0 |
| User j | 0 | 10 | 0 | 1 | 0 |
| User k | 1 | 1 | 0 | 0 | 0 |
| User l | 1 | 1 | 1 | 1 | 5 |
| Total | 3 | 12 | 3 | 7 | 5 |

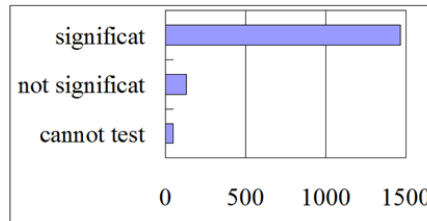


Fig. 5. Result of χ^2 test for common check-ins of users and their friends from the Gowalla dataset.

- Similarity by Relative Distance (DisSim): As discussed in [Ye *et al.*, 2010], most people will check-in to a POI following friends who live nearby. Based on this idea, the design of DisSim focuses on the distance between users and their friends. To this end, we must first identify users' living areas. We argue that most check-in activities will happen in a user's living area. Thus, we find each user's top- k frequently visited POIs, and treat the center of

these POIs as the *base-point*, as shown in Fig. 6. Accordingly, we formulate the DisSim of two users as

$$DisSim(u, f) = \begin{cases} 1 - \frac{Distance(u, f)}{\max_{f' \in F(u)} \{Distance(u, f')\}}, & \text{if user } u \text{ and } f \text{ are friends} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $Distance(*)$ indicates the Euclidean distance between two base-points and $F(u)$ indicates the set of user u 's friends.



Fig. 6. An example base-point.

4.1.2. Features from Preference-triggered Intentions. As discussed in the Introduction, we can view each user in the LBSN as a sensor for detecting the semantic tags of a POI in order to improve recommendations in an urban area. In many LBSN websites, there are two kinds of semantic tags, i.e., pre-defined categories and user-annotated highlights, as in Gowalla, shown in Fig. 7. The category tag is applied when the POI is created. Each place can have only one category tag, e.g., coffee, pizza. On the other hand, any user (even one who never visits the place) can arbitrarily annotate the highlight tag of a POI at any time. Indeed, the same tag can be applied to the same POI many times. Accordingly, the tag counts can be viewed to determine what type of activity occurs at a POI. Thus, the possibility that a tag t is annotated on a POI p can be formulated as

$$Possibility(t, p) = \frac{count(t, p)}{\sum_{t' \in T(p)} count(t', p)} \quad (6)$$

where $count(t, p)$ indicates the number of times t is annotated on p , and $T(p)$ indicates the set of tags of p . Taking Fig. 7 as an example, the possibility that the tag 'coffee' is annotated on a POI is

$$\frac{2}{2 + 10 + 88} = 0.02 \cdot$$

Intuitively, PreI features capture how the semantic labels of the POI match the user's personal preference. The personal preference reflects the degree to which the user likes particular semantic labels. Accordingly, given a user-POI pair (u, p) , the features extracted from PreI can be generally formulated as

$$PreI(u, p) = \sum_{t \in T(p)} (\text{Possibility}(t, p) \times \text{Personal Preference}(u, t)) \quad (7)$$

where $T(p)$ indicates the set of tags of POI p .

Significantly, we can observe that measuring a user's individual preference for a semantic tag is the key feature extracted from PreI. Intuitively, users' check-in histories can reflect their preference for types of POI. As a result, for each user, we aggregate the number of check-ins at POIs with the same tag to represent each user's personal preference. In the following, we extract two features to depict users' preferences.

- Preference in Category (CPref): Based on our observations from the LBSN dataset, users' check-in activity fluctuates. Some users frequently visit many places, whereas some users rarely check-in at all. Hence, to measure an individual user's personal preference for POIs, we need to normalize the aggregated number of check-ins based on the total number of user check-ins. Hence, we formulate user u 's personal preference for category tag t as

$$CPref(u, t) = \frac{\sum_{r \in C(t)} \text{checkins}(u, r)}{\#total\ check\ -\ ins\ of\ u} \quad (8)$$

where $C(t)$ indicates the set of POIs with category tag t . Using Tables I and II as an example, the check-in vector of user i is $\langle 1, 0, 2, 5, 0 \rangle$. Thus, user i 's personal preference for category tag A is

$$\frac{1 + 0 + 0}{1 + 0 + 2 + 5 + 0} = 0.125.$$



Fig. 7. An example of a POI's semantic tag.

Table II. Example category tags.

| POI ID | p1 | p2 | p3 | p4 | p5 |
|----------|----|----|----|----|----|
| Category | A | A | B | C | A |

Table III. Example of highlight tags.

| POI ID | p1 | p2 | p3 | p4 | p5 |
|-----------|------|---------|------|------|----|
| Highlight | a, b | b, c, d | a, d | a, c | g |

Note that, as mentioned above, people may annotate highlight tags on a POI and repeat the annotation many times. Based on this observation, to measure an individual user's personal preference of POIs, we cannot directly normalize the aggregated number of check-ins based on the total number of user check-ins. From Tables I and III, we can employ a user's total check-ins to normalize the aggregated number of check-ins, giving user i 's personal preference for highlight tag a as

$$\frac{1+2+5}{1+0+2+5+0} = 1.$$

Using the total check-ins to normalize the aggregated number of check-ins, it is clear that the total personal preference of a user will be greater than 1.0 and the scale of the total personal preference of a user will fluctuate considerably.

- Preference in Highlight (HPref): Based on the above observation and the availability of highlight tags, we use the summation of a user's total check-ins with each highlight tag to normalize their personal preference for POIs. To do so, given a set of highlight tags H , we formulate user u 's personal preference for highlight tag t as

$$\text{HPref}(u, t) = \frac{\sum_{r \in hl(t)} \text{Checkins}(u, r)}{\sum_{\forall t' \in H} \sum_{r \in hl(t')} \text{Checkins}(u, r)} \quad (9)$$

where $hl(t)$ indicates the set of POIs with highlight tag t . From Tables I and III, the check-in vector of user i is $\langle 1, 0, 2, 5, 0 \rangle$. Thus, user i 's personal preference for highlight tag a is

$$\frac{1+2+5}{(1+2+5) + (1+0) + (0+5) + (0+2) + (0)} = 0.5.$$

4.1.3. Features from Popularity-triggered Intentions. As discussed in the Introduction, check-in data is very sparse. This means that features extracted from the behavior motivated by both SI and PreI are not effective at recommending users' POIs, and his/her social circle never checks-in. We employ the total number of check-ins of a POI to make a maximum likelihood estimation of the popularity of POI. Thus, we can calculate the absolute proportion (AP) of check-ins to a POI to the total number of check-ins to all POIs. Formally, given a set of POIs P , the absolute proportion of check-ins for POI p is formulated as

$$\text{AP}(p) = \frac{\# \text{ check - ins of } p}{\sum_{p' \in P} (\# \text{ check - ins of } p')} \quad (10)$$

For example, if a POI has been visited 1,000 times and the total number of check-ins at all POIs is 30,000, the absolute proportion of check-ins is 1/30. However, the type of POI affects users' willingness to check-in. For example, people may buy a cup of coffee in a coffee shop every day, but rarely go to a French restaurant. Accordingly, we estimate the likelihood using conditional probability to find the Relative Popularity (RP) of a POI. RP indicates the probability that users visit the POIs that belong to category t . The formula for RP is as follows:

$$\text{RP}(p) = P(p | t \in T(p)) = \frac{\# \text{ total check - ins of } p}{\sum_{p' \in C(t)} (\# \text{ total check - ins of } p')} \quad (11)$$

where $T(p)$ indicates the set of tags of POI p and $C(t)$ indicates the set of POIs with category tag t . Take Table I and Table II as an example. The POIs with category tag A are $p1$, $p2$, and $p5$. The total check-ins for POIs $p1$, $p2$, and $p5$ are 3, 12, and 5, respectively. Thus, the popularity of POI $p1$ is

$$\frac{3}{3+12+5} = 0.15 \cdot$$

4.1.4. Summary of Features. We have proposed several features from SI, PreI, and PopI. An abstract description of these features is given in Table IV. Each intention contains two kinds of features. We can see that all the features from SI and PreI are associated with the user-POI pair, but those from PopI are only associated with the POIs. Therefore, the features from PopI are usually ineffective for POI recommendations, but they can deal with the problem of data sparsity (or the problem of cold starts). Besides, the features from PreI are generated from individual user check-ins. This always leads to the problem of data sparsity, because the number of POIs that a user visits is always much less than the total number of POIs. Therefore, we argue that the features extracted from SI can make up for this shortage.

Table IV. Summary of Features

| Intention | Feature | Notion | Description |
|--|--|----------|---|
| <i>Social-triggered Intentions</i> | Friends' check-ins for the POI weighted by common check-ins | CheckSim | The user's friends' check-ins to the POI in proportion to his friends' total check-ins, weighted by the user's and his friends' common check-ins. |
| | Friends' check-ins for the POI weighted by geographic distance | DisSim | User's friends' check-ins to the POI in proportion to his friends' total check-ins, weighted by the user's and his friends' geographic distance. |
| <i>Preference-triggered Intentions</i> | User's check-ins for the category term of the POI | CPref | User's check-ins for the category term of the POI in proportion to his total check-ins. |
| | User's check-ins for the highlight term of the POI | HPref | User's check-ins for the highlight terms of the POI in proportion to his total check-ins. |
| <i>Popularity-triggered Intentions</i> | Absolute proportion of check-ins | AP | Check-ins to the POI in proportion to total check-ins across the whole data set. |
| | Relative popularity | RP | Check-ins to the POI in proportion to total check-ins at POIs in the same category as that of the POI. |

4.2 User-POI Graph Construction

After the feature extraction phase, the features derived from SI, PreI, and PopI are used as weights for building a User-POI Graph (UPG). The UPG is a weighted complete bi-partite graph containing two partite sets, i.e., a user set and a POI set, as shown in Fig. 8. Because there are several features, there are various weighting functions for building the UPG. As the features derived from SI and PreI could not be associated with every user-POI pair, we use the features derived from PopI to replace them. This heuristic is based on the assumption that users would visit a popular POI, whereas they would not go to an area in which no POIs have been visited by his/her friends or satisfy his/her individual preferences. Accordingly we formally define several UPGs as follows:

Definition 4.1. User-POI Graph Weighted by Preference-triggered Intentions (UPG_{PreI}).

Given a user set U and a POI set P , the UPG weighted by PreI is a weighted complete bi-partite graph with the edge set $U \times P$. The weighting function is formulated as

$$w_{PreI}(u, p) = \begin{cases} PreI(u, p), & \text{if user } u \text{ has check in POI } p, \\ PopI(p), & \text{otherwise} \end{cases} \quad (12)$$

where $u \in U$ and $p \in P$.

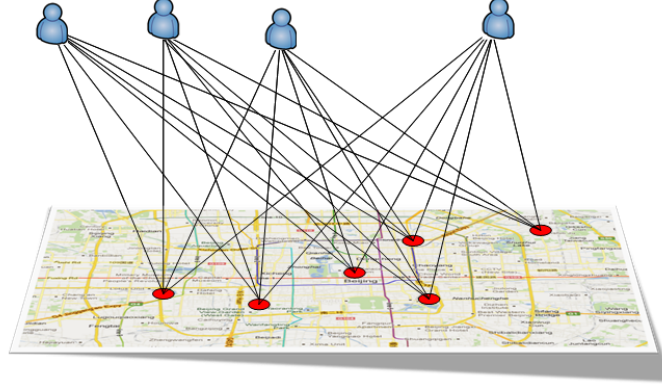


Fig. 8. Example of a user-POI graph.

There are four UPGs weighted by PreI, because there are two features extracted from PreI and two extracted from PopI. Intuitively, we have four combinations for the weighting function. These are denoted by $UPG_{PreI}\langle CPref, AP \rangle$, $UPG_{PreI}\langle CPref, RP \rangle$, $UPG_{PreI}\langle HPref, AP \rangle$, and $UPG_{PreI}\langle HPref, RP \rangle$.

Definition 4.2. User-POI Graph Weighted by Social-triggered Intentions (UPG_{SI}). Given a user set U and a POI set P , the UPG weighted by SI is a weighted complete bi-partite graph with the edge set $U \times P$. The weighting function is formulated as

$$w_{SI}(u, p) = \begin{cases} SI(u, p), & \text{if user } u \text{ has check in POI } p, \\ PopI(p), & \text{otherwise} \end{cases} \quad (13)$$

where $u \in U$ and $p \in P$.

Similarly, because there are two features extracted from SI and two from PopI, there are four kinds of UPG weighted by SI. Accordingly, we have four combinations for the weighting function. We denote these as $UPG_{SI}\langle CheckSim, AP \rangle$, $UPG_{SI}\langle CheckSim, RP \rangle$, $UPG_{SI}\langle DisSim, AP \rangle$, and $UPG_{SI}\langle DisSim, RP \rangle$.

4.3 Dynamic HITS-based Random Walk

Existing random walk approaches do not address the heterogeneous features discussed in the previous sections, instead using a straightforward aggregation of the extracted features, such as a linear combination. This leads to certain features dominating others. Given m users and n POIs, we build an $m \times n$ adjacency matrix M for the UPG. Formally, $M = [w_{ij}]$, $0 \leq i < m$; $0 \leq j < n$, where w_{ij} represents the weight of the edge linking the i th user and the j th POI. Formally, the random walk model applied to the UPG can be described as follows:

$$\begin{aligned} x_{POI}^{k+1} &= (\varepsilon M_{col}^T + (1 - \varepsilon)\delta_1)x_{user}^k \\ x_{user}^{k+1} &= (\varepsilon M_{row} + (1 - \varepsilon)\delta_2)x_{POI}^{k+1} \end{aligned} \quad (14)$$

where k indicates the number of iterations, M_{col} indicates the column stochastic matrix of M (computed by normalizing each column in M), M_{row} indicates the row stochastic matrix of M

(computed by normalizing each row in M), δ_1 indicates a matrix with all elements equal to $1/m$, δ_2 indicates a matrix with all elements equal to $1/n$, and ε indicates the “teleport probability,” which represents the probability of a random surfer teleporting from a POI node to a user node (respectively from a user node to a POI node) instead of following the links in the UPG.

```

Input: a set of the UPG
Output: relevance score of each user-POI pair
1   $R \leftarrow$  zero  $m \times n$  matrix //  $m$  users and  $n$  POIs
2  for each user  $i$ 
3     $u \leftarrow$  initial vector( $i$ ) // the  $i$ th value of vector is 1, others are 0.
4    for iteration  $< \tau$ 
5       $M \leftarrow$  random select(UPG)
6       $v \leftarrow (\varepsilon M_{col}^T + (1-\varepsilon) \delta_1)u$ 
7       $u \leftarrow (\varepsilon M_{row} + (1-\varepsilon) \delta_2)v$ 
8    end
9     $R[i] \leftarrow v^T$  //  $R[i]$  is the  $i$ th row of  $R$ 
10 end
11 Return  $R$ 

```

Fig. 9. Dynamic HITS-based Random Walk algorithm.

As we described earlier, there are a total of eight types of UPG. Therefore, we can randomly select one of these as the matrix M in each iteration. Based on this idea, we propose a novel Dynamic HITS-based Random Walk model, as shown in Fig. 9. First, the initial state of each user is obtained (see Lines 1 to 3 of Fig. 9). For each iteration, we randomly select one of the UPGs as the surfer teleporting between POI nodes and user nodes (see Lines 4 to 8 of Fig. 9). Finally, we can obtain the probability that each user achieves each POI v by the Dynamic HITS-based Random Walk (see Lines 9 to 11 of Fig. 9).

Example 1. Dynamic HITS-based Random Walk. Suppose that the adjacency matrix M , M_{col} , and M_{row} for a UPG are as follows:

$$M = \begin{bmatrix} 1 & 0 & 2 & 5 & 0 \\ 0 & 10 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 5 \end{bmatrix}, M_{col} = \begin{bmatrix} 1/3 & 0 & 2/3 & 5/7 & 0 \\ 0 & 10/12 & 0 & 1/7 & 0 \\ 1/3 & 1/12 & 0 & 0 & 0 \\ 1/3 & 1/12 & 1/3 & 1/7 & 5/5 \end{bmatrix}, M_{row} = \begin{bmatrix} 1/8 & 0 & 2/8 & 5/8 & 0 \\ 0 & 10/11 & 0 & 1/11 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/9 & 1/9 & 1/9 & 1/9 & 5/9 \end{bmatrix}$$

If we want to calculate the relevance of the first user to other users, we initialize x_{user} as $(1, 0, 0, 0)^T$, and then use the power iteration shown in Equation (14) to calculate the vectors x_{user}^k . If we set the number of iterations k to 2 and the teleport probability ε to 1, the relevance of the first user to other users can be calculated as follows:

$$M_{row} M_{col}^T x_{user} = \begin{bmatrix} 1/8 & 0 & 2/8 & 5/8 & 0 \\ 0 & 10/11 & 0 & 1/11 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/9 & 1/9 & 1/9 & 1/9 & 5/9 \end{bmatrix} \begin{bmatrix} 1/3 & 0 & 1/3 & 1/3 \\ 0 & 10/12 & 1/12 & 1/12 \\ 2/3 & 0 & 0 & 1/3 \\ 5/7 & 1/7 & 0 & 1/7 \\ 0 & 0 & 0 & 5/5 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 55/84 \\ 5/77 \\ 1/6 \\ 4/21 \end{bmatrix}$$

Accordingly, we find the relevance probabilities for the first user to the second, third, and fourth users to be $5/77$, $1/6$, and $4/21$, respectively.

5. POI RECOMMENDATION

After the relevance learning step, we must estimate the relevance between users and POIs. As the POIs in urban areas are very crowded, there will be too many POIs in the recommendation list. Although we can use some conditions from the users' environment, such as geographic distance, to filter the possible POIs, there may still be too many for users to handle. We thus use the relevance results to rank the POIs in the recommendation list. As demonstrated in Fig. 1, the POI recommendation phase is similar to the process of information retrieval. All users are considered as queries to search for their top- k nearest POIs. After finding the top- k nearest POIs, the matrix recording the relevance between users and POIs is used to rank the list of POIs. For instance, using u_k as a query, we retrieve the ten nearest POIs based on the geographical distance between u_k 's current position and the positions of the POIs. The estimated relevance vector of the search results is then formulated according to the user-POI relevance matrix. Based on the estimated relevance vector, we can rank the POIs from high to low relevance.

6. EXPERIMENTAL EVALUATIONS

In this section, we present the results from a series of experiments to evaluate the performance of UPOI-Walk using Gowalla and EveryTrail datasets. All the experiments are implemented in Java JDK 1.6 on an Intel Xeon CPU W3520 2.67 GHz machine with 24 GB of memory running Microsoft Windows 7. We first describe the preparation of the datasets, and then introduce the evaluation methodology. Finally, we present and discuss our experimental results.

6.1 Datasets

6.1.1 Gowalla Dataset. The Gowalla dataset is a check-in dataset collected from the Gowalla website. The dataset contains 1,964,919 POIs, 18,159 users, 5,341,191 check-ins, and 392,246 friend links. We normalize the check-ins for each user, and use the normalized check-ins as the ground truth, which we refer to as the "relevance ground-truth." We normalize the check-in values in the range [0, 5]. The process of normalization is described by Equation (15).

$$\text{Relevance Score} = \begin{cases} 3 + \frac{x - \text{avg}}{\text{max} - \text{avg}} \times 2, & \text{if } x > \text{avg} \\ 3, & \text{if } x = \text{avg} \\ 3 - \frac{x - \text{avg}}{\text{min} - \text{avg}} \times 2, & \text{if } x < \text{avg} \end{cases} \quad (15)$$

where x indicates the real check-in number of a user. "min," "max," and "avg" indicate the smallest, largest, and average check-in number of a user, respectively.

6.1.2 EveryTrail Dataset. EveryTrail is a trip-sharing and social networking website on which users can upload, share, and find trips. The data includes GPS logs and photos within a trip. Users can also label an activity in a trip. The EveryTrail website provides a public API to allow other applications to integrate with their service, although some functionality in the API is broken. For this reason, we crawl web pages in support of the API in order to obtain all necessary data. However, the trajectory data is very different from check-in data. We must preprocess the trajectory data to transform it into suitable check-in behavior. Accordingly, we generate locations from the trajectory dataset. Although there are many techniques

to discover POIs from GPS data, such as grid-based approaches [Giannotti *et al.*, 2006; Monreale *et al.*, 2009] and density-based approaches [Jeung *et al.*, 2008; Mamoulis *et al.*, 2004], we want to know where users are checking in. To this end, the “stay” parameter is important for detecting POIs. Obviously, these works do not address the “stay” aspect. As a result, for each user, we adopt the notion of *stay locations* [Zheng *et al.*, 2009] to represent users’ check-in behavior, as shown in Fig. 10. To discover stay locations, we first detect the regions, called stay points, in which a user stayed, i.e., *s1* and *s2* in Fig. 10. We then cluster all detected stay points to form stay locations, i.e., *location2* and *location5* in Fig. 10. As shown in Fig. 10, the user moving along the trajectory could be viewed as checking in to *location2* and *location5*. We directly use the activity tags of the trajectory to annotate the locations that it passes. As the tags are provided by the user, and a location may be annotated with the same tag several times, we can treat this as a “highlight tag.” In other words, the PreI features can also be extracted from the EveryTrail dataset. After completing this process, we have 24,889 locations, 1,133 users, 843,009 check-ins, and 102,861 friend links.

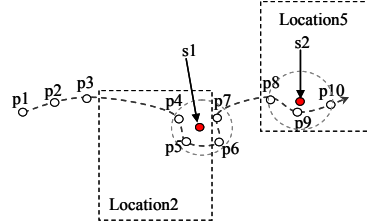


Fig. 10. Example of a stay location.

6.2 Evaluation Methodology

UPOI-Walk is based on the ranking of POI scores, and can thus be viewed as an information retrieval system with users as the query terms. Therefore, we employ the popular measurement of Normalized Discounted Cumulative Gain (NDCG) [Manning *et al.*, 2008] to evaluate the list of recommended POIs. NDCG is commonly used in information retrieval to measure the performance of search engines. A higher NDCG value for a list of search results indicates that highly relevant items have appeared earlier (with higher ranks) in the result list. For each list of recommended POIs, we can obtain a score list, where the scores are provided by the ground truth. Such a list is called the relevance vector. For example, suppose that the prediction model estimates the relevance scores of POIs $F1$, $F2$, $F3$, and $F4$ to be 4, 3, 2, and 1, respectively. Hence, the POIs will be ordered as $\langle F1, F2, F3, F4 \rangle$ by the recommender. Suppose that the relevance score of the ground truth for POI list $\langle F1, F2, F3, F4 \rangle$ is $G = \langle 2, 3, 0, 1 \rangle$. That is, the relevance scores of $F1$ and $F2$ are 2 and 3, respectively. The DCG of relevance vector G is computed by Equation (16). The premise of DCG is that highly relevant documents appearing lower in a search result list should be penalized, as the graded relevance value is reduced logarithmically in proportion to the position of the result. Here, the parameter b controls where we start to reduce the relevance value. For example, if the relevance vector is $\langle 2, 3, 0, 1 \rangle$ and b is set as 3, the DCG [Hu *et al.*, 2007] is $2+3+(0/\log_3 3)+(1/\log_3 4)$. (In our experiments, $b = 2$.)

$$DCG[i] = \begin{cases} G[i], & \text{if } i = 1 \\ DCG[i-1] + G[i], & \text{if } i < b \\ DCG[i-1] + \frac{G[i]}{\log_b i}, & \text{if } i \geq b \end{cases} \quad (16)$$

In particular, $NDCG@p$, measures the relevance of the top p , as shown in Equation (17).

$$NDCG@p = \frac{DCG[p]}{IDCG[p]} \quad (17)$$

where $IDCG[p]$ indicates the $DCG[p]$ value of the ideal ranking list. For example, given a ranking list of five items with relevances $\langle 4, 1, 3, 1, 1 \rangle$, the ideal ranking list is $\langle 4, 3, 1, 1, 1 \rangle$. NDCG scores range from 0 to 1. The higher the NDCG score, the better the ranking result list. In the above example, the $NDCG@5$ is

$$NDCG@5 = \frac{4 + \frac{1}{\log_2 2} + \frac{3}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5}}{4 + \frac{3}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} + \frac{1}{\log_2 5}} = 0.913785$$

However, NDCG is not comprehensive, because it only focuses on ranking performance, avoiding the absolute difference between estimated relevance and the ground truth. For example, given a ranking list of five items with estimated relevances $\langle 5, 4, 3, 1, 1 \rangle$, the ground truth for the relevance of these five items is $\langle 4, 3, 2, 2, 2 \rangle$. We can see that the NDCG will be 1.0, implying that the effectiveness of the recommender is good. Besides the ranking performance, we also want to examine how close our predictions are to the eventual outcomes. Therefore, we employ the Mean Absolute Error (MAE) to evaluate the list of recommended POIs.

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \quad (18)$$

where f_i indicates the estimated relevance scores of POI i and y_i indicates the ground truth of relevance scores. The lower the MAE score, the fewer the number of errors. In the above example, the MAE is

$$\frac{1}{5} \times (|5-4| + |4-3| + |3-2| + |1-2| + |1-2|) = 1.0 \cdot$$

6.3 Experimental Results and Discussion

The experiments can be divided into internal and external experiments. For the internal experiments, we first examine the optimal setting for the teleport probability (ϵ) in terms of NDCG and MAE. We then compare the effectiveness of every feature extracted from all three types of intention according to these measures. For the external experiments, we compare our method with the following five state-of-the-art approaches in terms of NDCG and MAE.

- CF-based models [Ye *et al.*, 2011b]. Ye *et al.* proposed a CF-based POI recommendation framework that fuses user preference influence, social influence, and geographic influence to infer the check-in probability of a given user at a POI. Therefore, this approach contains three sub-routines, which we call *CF-based (user preference influence)*, *CF-based (social influence)*, and *CF-based (geographic influence)*.

- CF-based models [Zheng *et al.*, 2010]. Zheng *et al.* proposed a CF-based approach in which users can employ an activity term (semantic tag), such as hiking, as a query term to request the location recommendation service.
- TrustWalker [Jamali and Ester, 2009]. The TrustWalker model combines a random walk model with a trust-based CF approach to predict the ratings of items. The trust-based CF approach uses trust values and user-to-item ratings to predict the item ratings. The trust value is the user-to-user rating.
- HITS-based Random Walk (denoted as linear combination). We use the equal weight linear combination of all features we mentioned in Section 4.1. to build a user-POI graph and performs HITS-based Random Walk on it.
- HITS-based Random Walk [Cao *et al.*, 2010]. Cao *et al.* performs a HITS-based Random Walk on the user-POI graph and POI-POI graph. They utilized the number of check-ins as the weight of edge of user-POI graph and the number of consequent check-ins as the weight of edge of POI-POI graph.
- UPOI-Mine [Ying *et al.*, 2012] (denoted as M5Prime). This approach is our previous work, which uses the features stated in Section 4.1 to learn a regression-tree model, M5Prime, for relevance estimation.

6.3.1. Tuning Parameters. Because the teleport probability (ε) plays a crucial role in the performance of the HITS-based Random Walk, we vary it in our Dynamic HITS-based Random Walk. This allows us to understand the weighting required to achieve the optimal Dynamic HITS-based Random Walk performance. As shown in Fig. 11, the optimal performance occurs for a teleport probability $\varepsilon = 0.4$, except for NDCG@5 in the EveryTrail dataset. In other words, a lot of behavior is similar to “randomly” moving. This is because the check-in data is very sparse, as mentioned earlier, and thus a lot of behavior recorded in the data could be viewed as “random” movement. If we forced some heuristic to represent users’ behavior, the performance of the model would suffer. Accordingly, we set the teleport probability $\varepsilon = 0.4$ in the following experiments.

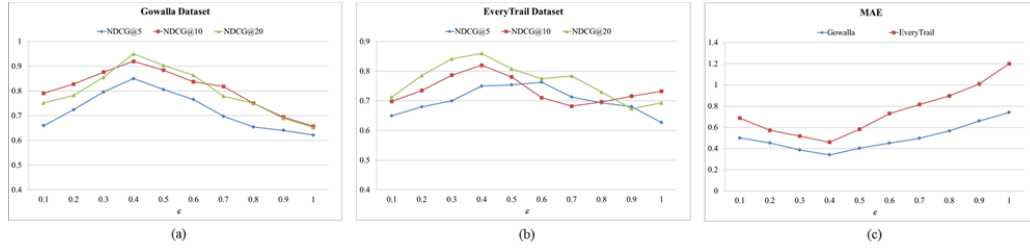


Fig. 11. Comparison of various teleport probabilities.

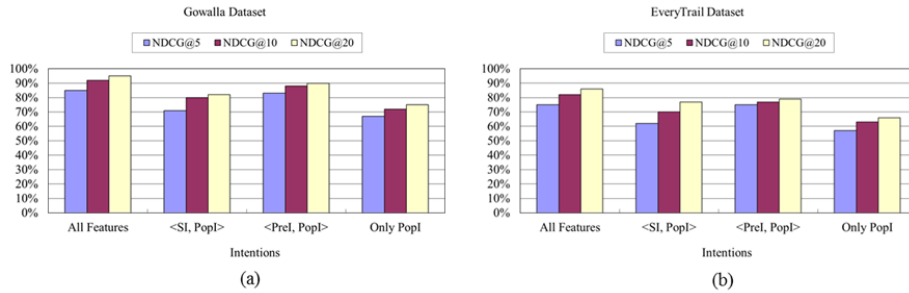


Fig. 12. Comparison of different intentions in terms of NDCG.

6.3.2. Comparison of Various Features in terms of NDCG. This experiment evaluates the performance of each feature in the proposed UPOI-Walk in terms of NDCG@5, NDCG@10, and NDCG@20. Fig. 12 shows the results. We observe that the effectiveness of features from PreI is better than that of features from SI. This result shows that users' self-preference motivates their check-in behavior more strongly than the preferences of their friends. However, we should still use features from PopI and SI to improve recommendations, because the effectiveness of using features from all intentions outcores the other combinations.

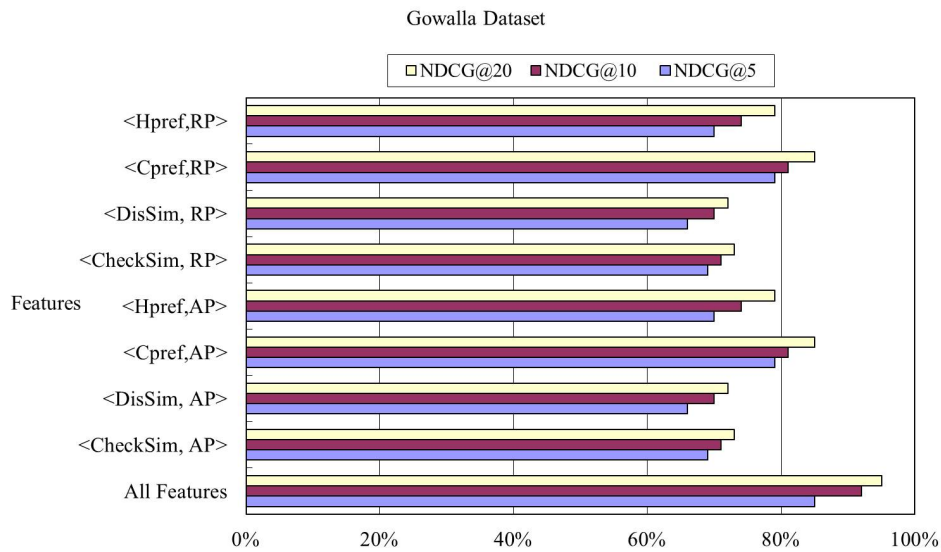


Fig. 13. Comparison of different features in terms of NDCG for the Gowalla dataset.

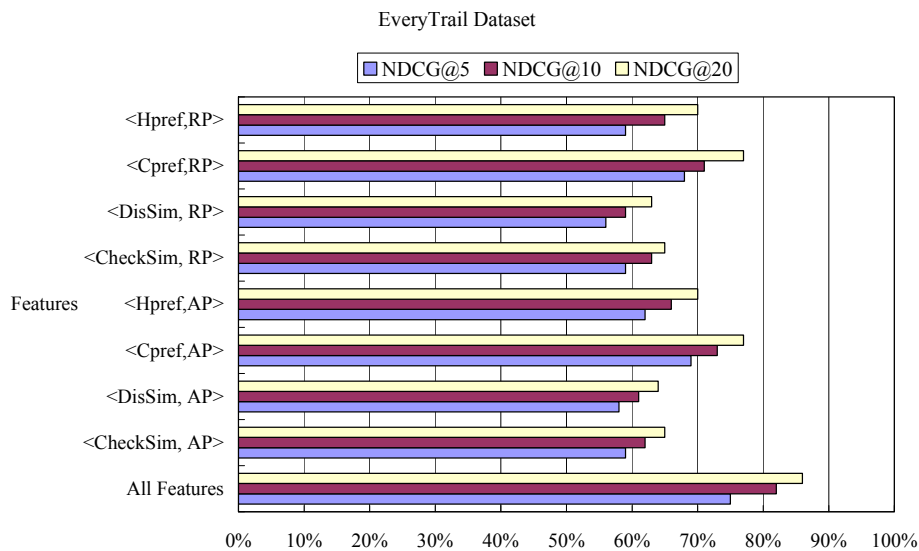


Fig. 14. Comparison of different features in terms of NDCG for the EveryTrail dataset.

In Figs. 13 and 14, we compare all features in terms of NDCG@5, NDCG@10, and NDCG@20 for the Gowalla and EveryTrail datasets, respectively. We observe that the CPref feature outperforms all others, because the POI category is able to represent the users' preferences. In the EveryTrail dataset, because the trajectory tags are provided by users, the PreI features reflect the users' preferences completely. As a result, CPref and HPref outperform the other features.

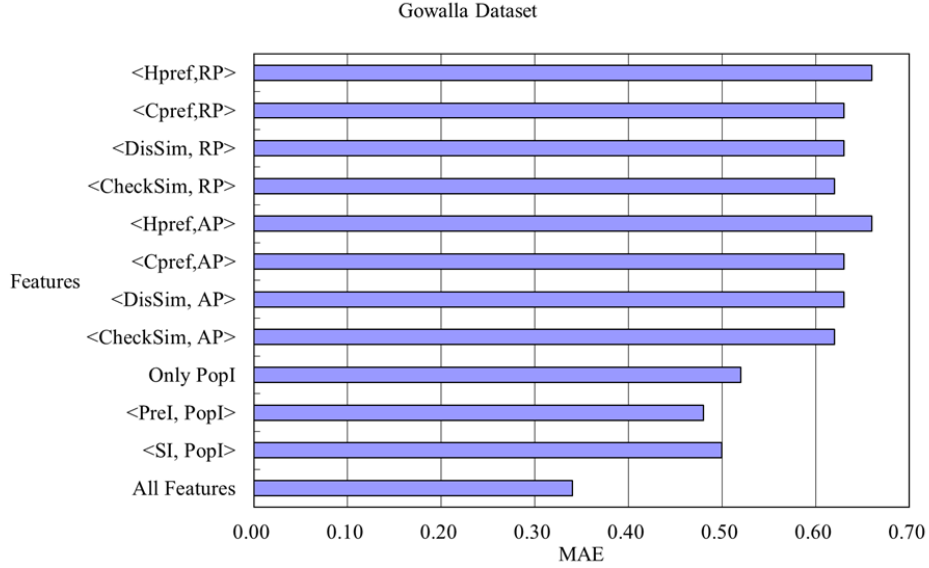


Fig. 15. Comparison of different features in terms of MAE.

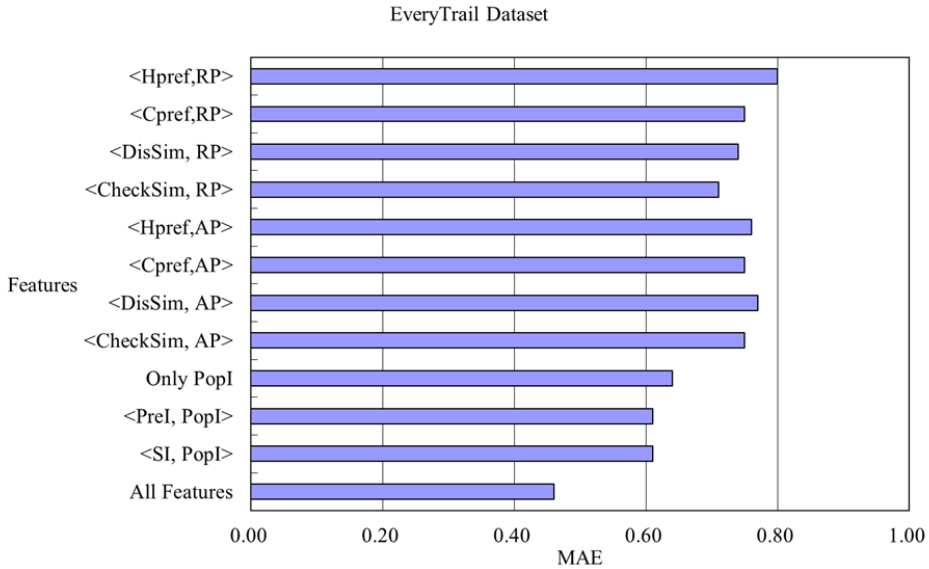


Fig. 16. Comparison of different features in terms of MAE.

6.3.3. Comparison of Various Features in terms of MAE. This experiment evaluates the performance of each feature in the proposed UPOI-Walk in terms of MAE. Figs. 15 and 16 both show that the MAE is smallest when combining all features, but the

features from PreI are not significantly effective. This is because the three intentions are complementary when using a random walk model. If we try to utilize only one feature to perform the random walk, the model will over-fit that aspect. This means that the model can only reflect partial properties of the dataset, damaging the accuracy of the model in some cases.

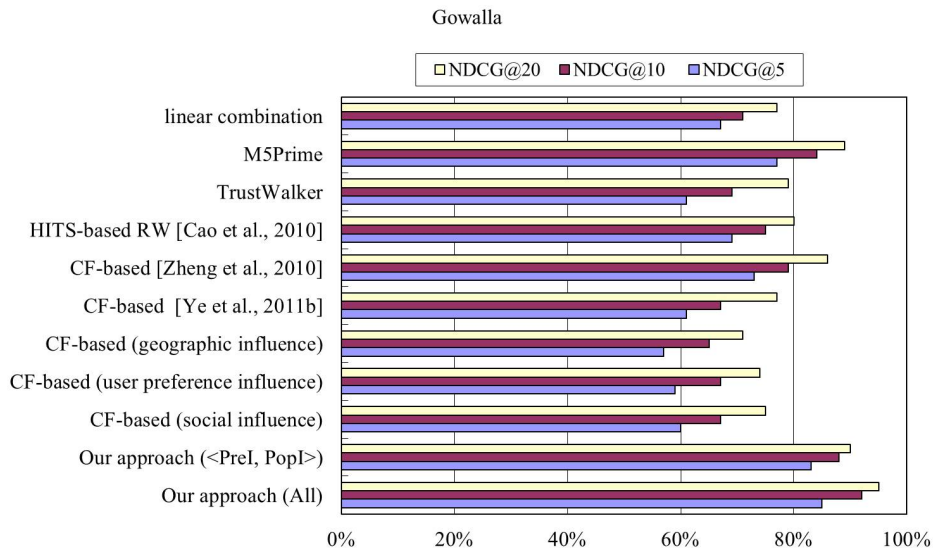


Fig. 17. Comparison of various models in terms of NDCG with the Gowalla dataset.

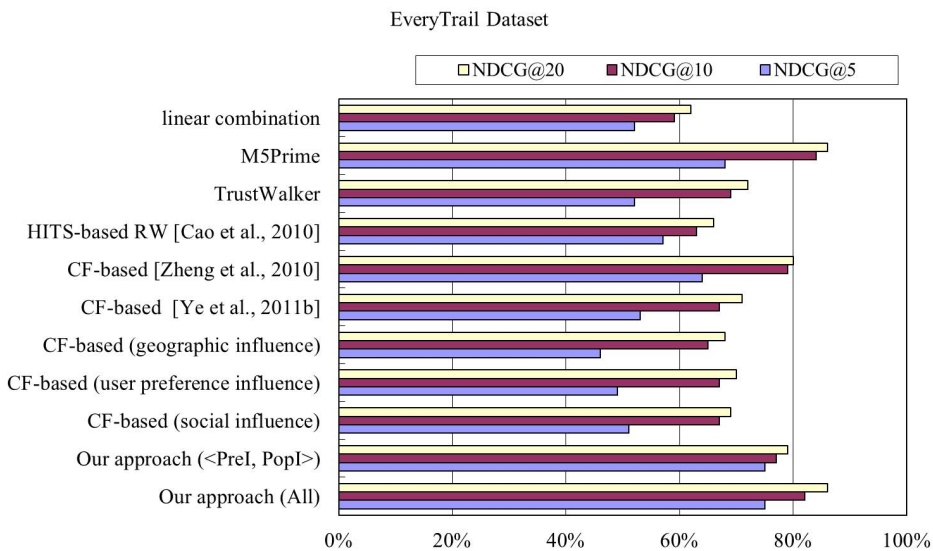


Fig. 18. Comparison of various models in terms of NDCG with the EveryTrail dataset.

6.3.4. Comparison of existing recommendation models. This experiment compares the performance of our proposed UPOI-Walk with that of HITS-based Random Walk[Cao *et al.*, 2010], TrustWalker [Jamali and Ester, 2009], UPOI-Mine [Ying *et al.*, 2012],

and two CF-based models [Zheng *et al.*, 2010; Ye *et al.*, 2011b] in terms of NDCG@5, NDCG@10, NDCG@20, and MAE. Figs. 17 and 18 show that UPOI-Walk outperforms other existing works in terms of NDCG. This is because we consider check-ins motivated by PreI, whereas other methods do not. Although one of the CF-based models is described as being influenced by “preference,” it directly utilizes users’ common check-ins to estimate their POI preferences, instead of the preference for a semantic tag.

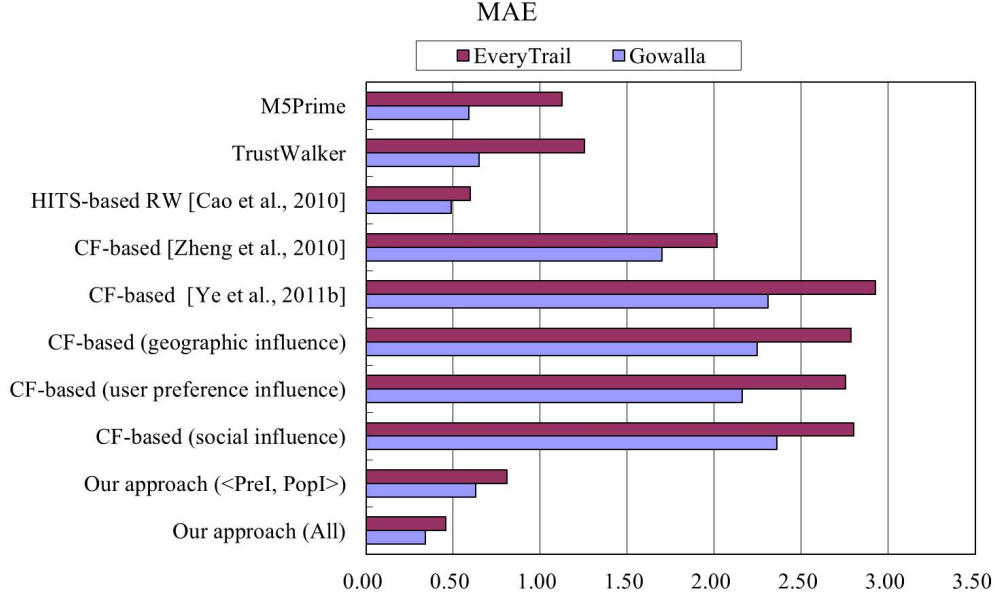


Fig. 19. Comparison of various models in terms of MAE.

Besides, we also can observe that the linear combination does NOT outperform our UPOI-Walk. Although the linear combination and our UPOI-Walk use the same features, features extracted from PopI might derive more influence for recommendation. Therefore, the influence of other features is diluted by the influence of features extracted from PopI. We also found that the MAE of CF-based POI is large, as shown in Fig. 19. Because of the sparsity of LBSN data, considering common check-ins for estimating users’ similarity will lead to many zero scores in the user-POI matrix. The reason for these large MAE values is that the zero scores are far from users’ actual relevance scores. On the contrary, UPOI-Walk, HITS-based Random Walk and TrustWalker evaluate the relevance between a user and a POI by using the check-ins of other users who have no common check-in with the user.

6.3.5. Scalability in Dynamic HITS-based Random Walk. Generally, Random walk algorithms require many matrix multiplications in the iterative process. This makes them very time-consuming. To examine the scalability of our method, we measured the learning time using the dataset crawled from Gowalla website under different number of POIs and number of users. The results are shown in Fig. 20. We can observe that the learning time increased exponentially as the size of the dataset increased. Fortunately, the relevance learning can be processed offline. Besides, according to the observation of [Cho *et al.*, 2011], people rarely make long-distance travel. Thus, the data could be divided by city, and we could parallelize our Dynamic

HITS-based Random Walk by performing relevance learning on each piece of data. The Gowalla dataset consists of POIs for the whole of New York City, so it is reasonable that the learning step takes about 4 hours.

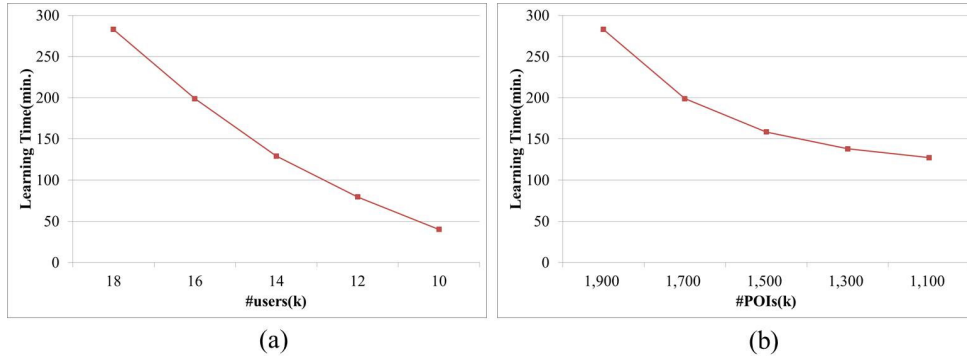


Fig. 20. Scalability in Dynamic HITS-based Random Walk

6.4 Discussion of Experimental Results

Through the above series of experiments, we have shown that the PreI features are significantly more effective than those of SI. This is because many activities are actually motivated by the users themselves, although most check-in behavior is affected by their friends. For example, people usually change their eating habits, although they seldom side with their friends. Accordingly, UPOI-Walk and UPOI-Mine outperform other existing recommenders, which focus only on the check-in behavior motivated by SI. Furthermore, we can also observe that UPOI-Walk outperforms UPOI-Mine, especially in terms of MAE. The reason for this is that our proposed Dynamic HITS-based Random Walk randomly selects different types of UPG for the random walk process. Hence, our model comprehensively considers all intentions used for feature extraction. Therefore, the Dynamic HITS-Based Random Walk model is very effective for dealing with heterogeneous data problems.

7. CONCLUSIONS

In this paper, we have proposed a novel approach named *Urban POI-Walk (UPOI-Walk)* for recommending interesting urban POIs by mining users' preferences. We also tackled the problem of mining user check-in behavior in an urban computing environment, which is a crucial prerequisite for effective POI recommendations in urban areas. The core task of POI recommendation in urban areas is conveniently transformed to the problem of relevance score prediction. We evaluated the relevance score of each user-POI pair by training a random walk model. In the proposed *UPOI-Walk*, we have explored i) *Social-triggered Intentions*, ii) *Preference-triggered Intentions*, and iii) *Popularity-triggered Intentions* by exploiting LBSN data to extract descriptive features. To the best of our knowledge, this is the first work on urban POI recommendation that considers user check-in behaviors motivated by Social-, Preference-, and Popularity-triggered Intentions in LBSN data. Furthermore, to deal with the heterogeneous feature problem, we proposed a novel random walk model, called *Dynamic HITS-Based Random Walk*, to learn the user-POI relevance. Through a series of experiments using two real datasets, Gowalla and EveryTrail, we have validated the proposed *UPOI-Walk* and shown that it has excellent performance under various conditions. In future work, we plan to design more sophisticated

methods to enhance the quality of our proposed *Dynamic HITS-Based Random Walk* for various urban computing service applications.

REFERENCES

- Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R. and Pinkerton, M. 1997. Cyberguide: a mobile context-aware tour guide. In *Wireless Network*, 3(1997), 421-433.
- Berjani, B. and Strufe, T. 2011. A Recommendation System for spots in Location-Based Online Social Network. In *Proceedings of the 4th Workshop on Social Network Systems* Article No. 4.
- Bao, J., Zheng, Y. and Mokbel, M. F. 2012. Location-based and Preference-Aware Recommendation Using Sparse Geo-Social Networking Data. In *Proceedings of ACM SIGSPATIAL GIS 2012*
- Cao, X., Cong, G. and Jensen, C. S. 2010. Mining Significant Semantic Locations From GPS Data. *Proceedings of VLDB 2010*.
- Cho, E., Myers, S. A. and Leskovec, J. 2011. Friendship and Mobility: User Movement In Location-Based Social Networks. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.
- Debnath, S., Ganguly, N. and Mitra, P. 2008. Feature Weighting in Content Based Recommendation System Using Social Network Analysis. In *Proceedings of WWW*, 1041-1042.
- Etemad-Shahidi, A. and Mahjoobi, J. 2009. Comparison between M5' Model Tree and Neural Networks for Prediction of Significant Wave Height in Lake Superior. *Ocean Engineering*, 36, 1175–1181.
- Giannotti, F., Nanni, M., and Pedreschi, D. 2006. Efficient mining of temporally annotated sequences. In *Proceedings of the 6th SIAM International Conference on Data Mining*, Bethesda, MD, April, 2006.
- Hu, J., Zeng, H.-J., Li, H., Niu, C. and Chen, Z. 2007. Demographic Prediction based on User's Browsing Behavior. In *Proceedings of ACM WWW'07*.
- Horozov, T., Narasimhan, N. and Vasudevan, V. 2006. Using Location for Personalized POI Recommendations in Mobile Environments. In *Proceedings of SAINT*, 124-129.
- Jamali, M. and Ester, M. 2009. TrustWalker: A Random Walk Model for Combining Trust-based and Item-based Recommendation. In *Proceedings of KDD*, 397-406.
- Jeung, H., Liu, Q., Shen, H. T. and Zhou, X. 2008. A Hybrid Prediction Model for Moving Objects. In *Proceedings of ICDE'08*.
- Leung, K. W.-T., Lee, D. L. and Lee, W.-C. 2011. CLR: A Collaborative Location Recommendation Framework based on Co-Clustering. In *Proceedings of SIGIR*, 305-314.
- Levandovski, J.J., Sarwat, M., Eldawy, A. and Mokbel, M.F. 2012. LARS: A Location-Aware Recommender System. In *Proceedings of ICDE'12*
- Lu, E. H.-C., Lee, W.-C. and Tseng, V. S. 2012. A Framework for Personal Mobile Commerce Pattern Mining and Prediction. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 24(5), 769-782.
- Nikos Mamoulis, Huiping Cao, George Kollios, Marios Hadjieleftheriou, Yufei Tao, David W. Cheung. 2004. Mining, Indexing, and Querying Historical Spatio-temporal Data. *KDD 2004*.
- Manning, D., Raghavan, P. and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- Massa, P. and Avesani, P. 2007. Trust-Aware Recommender Systems, In *Proceedings of RecSys*, 17-24.
- Menzies, T., Stefano, J. S. D. and Chapman, M. 2003. Learning Early Lifecycle IV&V Quality Indicators. In *Proceedings of the Ninth International Software Metrics Symposium*.
- Monreale, A., Pinelli, F., Trasarti, R. and Giannotti, F. 2009. WhereNext: a Location Predictor on Trajectory Pattern Mining. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.
- Ono, C., Kurokawa, M., Motomura, Y. and Asoh, H. 2007. A Context-Aware Movie Preference Model Using a Bayesian Network for Recommendation and Promotion. In *Proceedings of UM*, 247-257.
- Park, H., Hong, H. and Cho, B. 2007. Location-based recommendation system using Bayesian user's preference model in mobile devices. In *Proc. of Ubiquitous Intelligence and Computing (Hong Kong, 2007)*. Springer Press: 1130-1139.
- Scellato, S., Noulas, A. and Mascolo, C. 2011. Exploiting Place Features in Link Prediction on Location-based Social Networks, In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.
- Simon, R. and Fröhlich, P. 2007. A mobile application framework for the geospatial Web. In *Proc. of the 16th Intl. Conf. on World Wide Web (Canada, 2007)*. ACM Press: 381-390.
- Spertus, E., Sahami, M. and Buyukkokten, O. 2005. Evaluating Similarity Measures: A Large-Scale Study in The Orkut Social Network. In *Proceedings of KDD*, 678-684.
- Takeuchi, Y. and Sugimoto, M. 2006. CityVoyager: an outdoor recommendation system based on user location history. In *Proc. of Ubiquitous Intelligence and Computing (Berlin Germany, 2006)*, Springer Press: 625-636.
- Ye, M., Janowicz, K., Mulligann, C. and Lee, W.-C. 2012a. What you are is when you are: the temporal

- dimension of feature types in location-based social networks. *ACM SIGSPATIAL* 2012.
- Ye, M., Liu, X. and Lee, W.C. 2012b. Exploring social influence for recommendation: a generative model approach. *ACM SIGIR* 2012.
- Ye, M., Shou, D., Lee, W.C., Yin, P. and Janowicz, K. 2011a. On the semantic annotation of places in location-based social networks. *ACM KDD* 2011.
- Ye, M., Yin, P., Lee, W.-C. and Lee, D.-L. 2011b. Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation. In *Proceedings of SIGIR*, 1046-1054.
- Ye, M., Yin, P., Lee, W.-C. and Lee, D.-L. 2010. Location recommendation for location-based social networks. In *Proceedings of GIS'10*.
- Ying, J. J.-C., Lu E. H.-C., KUO, W.-N. and Tseng, V. S. 2012. Urban Point-of-Interest Recommendation by Mining User Check-in Behaviors. in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing (UrbComp 2012)*, August 12, 2012, Beijing, China.
- Yuan, J., Zheng, Y. and Xie, X. 2011. Urban Computing with Taxicabs. In *Proceedings of 13th ACM International Conference on Ubiquitous Computing (UbiComp)*.
- Yuan, J., Zheng, Y. and Xie, X. 2012. Discovering Regions of Different Functions in a City Using Human Mobility and POIs. In *Proceedings of 18th SIGKDD conference on Knowledge Discovery and Data Mining (KDD)*.
- Zheng, Y. 2012. Tutorial on Location-Based Social Networks. In *Proceedings of ACM WWW'12*.
- Zheng, Y., Zhang, L., Xie, X. and Ma, W. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. of the 18th Intl. Conf. on World Wide Web (Madrid Spain, 2009)*, ACM Press: 791-800.
- Zheng, Y., and Zhou, X. 2011. *Computing with Spatial Trajectories*, Springer Publishing Company, Incorporated, 2011
- Zheng, V. W., Cao, B., Zheng, Y., Xie, X. and Yang, Q. 2010. Collaborative Filtering Meets Mobile Recommendation: A User-centered Approach, in *AAAI 2010*, ACM. 2010

Received October 2012; revised March 2013; accepted August 2013