

Graph-based Point-of-interest Recommendation with Geographical and Temporal Influences

Quan Yuan, Gao Cong, Aixin Sun
School of Computer Engineering, Nanyang Technological University, Singapore 639798
{qyuan1@e., gaocong@, axsun@}ntu.edu.sg

ABSTRACT

The availability of user check-in data in large volume from the rapid growing location-based social networks (LBSNs) enables a number of important location-aware services. Point-of-interest (POI) recommendation is one of such services, which is to recommend POIs that users have not visited before. It has been observed that: (i) users tend to visit nearby places, and (ii) users tend to visit different places in different time slots, and in the same time slot, users tend to periodically visit the same places. For example, users usually visit a restaurant during lunch hours, and visit a pub at night. In this paper, we focus on the problem of *time-aware POI recommendation*, which aims at recommending a list of POIs for a user to visit at a given time. To exploit both *geographical and temporal influences* in time-aware POI recommendation, we propose the Geographical-Temporal influences Aware Graph (GTAG) to model check-in records, geographical influence and temporal influence. For effective and efficient recommendation based on GTAG, we develop a preference propagation algorithm named *Breadth-first Preference Propagation* (BPP). The algorithm follows a relaxed breadth-first search strategy, and returns recommendation results within at most 6 propagation steps. Our experimental results on two real-world datasets show that the proposed graph-based approach outperforms state-of-the-art POI recommendation methods substantially.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information Filtering

Keywords

Recommendation; Point-of-interest; Location-based Social Networks; Preference Propagation; Spatio-Temporal

1. INTRODUCTION

Location-based social networks (LBSNs), such as Foursquare and Facebook Places, have become increasingly popular recently. They provide a platform for millions of users to share their geographical locations and experiences through *check-ins*. Each check-

Permission to make digital or hard copies of all or part 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 bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright 2014 ACM 978-1-4503-2598-1/14/11 ...\$15.00.
<http://dx.doi.org/10.1145/2661829.2661983>.

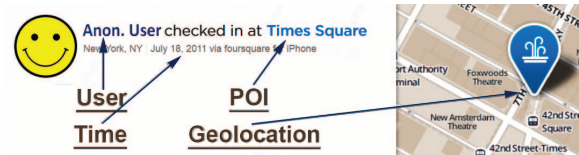


Figure 1: An example check-in

in represents a user's visit to a point-of-interest (POI), such as a restaurant, a sightseeing site, etc., at a specific visiting time. Figure 1 gives an example check-in: a user named *Anon. User* visited *Times Square* on *July 18, 2011*. As of January 2014, Foursquare accumulated over 5 billion check-ins made by 45 million users.

The availability of user check-in data in large volume offers the opportunity to design appealing services to facilitate users' travels and social interactions. POI recommendation is one of such services, which aims at recommending POIs to users who have not visited them before. Note that, it is common to have thousands of POIs even in a small city, and a user may have visited only a small portion of them. POI recommendation can help both residents and visitors to explore new interesting places in a city. On the other hand, POI recommendation certainly benefits the business owners of POIs for attracting more customers.

In this paper, we focus on the problem of *time-aware POI recommendation*, which is to recommend POIs for a target user to visit at a specific time slot in a day (target time). The recommendation results are time-aware due to the observation that a user tends to go to different places at different time slots. For example, a person is more likely to go to a pub rather than a restaurant at midnight.

From historical check-in records obtained from Foursquare or other sources, two observations have been made in existing studies related to POI recommendation. First, many users tend to visit nearby places [7, 30, 32]. We refer to this observation as the *geographical influence* on users' daily activities. Second, many users tend to visit different places at different time slots and periodically visit the same places in the same time slot [7, 17, 32]. For example, users usually go to work places in the morning and have dinner at restaurants in the early evening. We refer to this observation as the *temporal influence* on users' daily activities.

The geographical influence suggests that the chance of a user visiting a POI will be higher if this POI is closer to the user's previously visited POIs. In other words, a POI recommendation algorithm should consider geographical influence and rank such POIs higher. On the other hand, the temporal influence suggests that a POI recommendation algorithm shall consider user mobility pattern and recommend the right types of places based on the time a user plans to visit. For example, when we recommend POIs for a

user to visit during lunch hour, a restaurant should be given a higher priority than a pub.

To exploit both geographical and temporal influences for time-aware POI recommendation, we propose a graph-based approach. In our approach, a graph named *Geographical-Temporal influences Aware Graph* (GTAG) is constructed using all users' check-in records. GTAG consists of three types of nodes (user node, POI node, and session node) and two types of links (check-in link and POI link). Each user is represented by a user node that connects to a set of session nodes of the user, each of which represents a time slot (*e.g.*, an hour in a day). A session node of a user is connected to a POI node if the user of the session node visits the POI in the corresponding time slot. A pair of edges (between user node and session node, and between session node and POI node, respectively) form a check-in link, which represents a check-in record of the user. Two POI nodes are connected through a POI link if they are located geographically near each other. Hence, GTAG embeds both the temporal and geographical information from the check-in records. Introducing the session nodes for each user will make the data more sparse, which is a challenge for recommendation. To alleviate the data sparsity problem, we propose to compute the importance of each session, and the sessions close to the target session are assigned larger weights, since the behaviors of a user in two close time slots are similar.

Apart from effectiveness, efficiency is also an important consideration in designing recommendation algorithms, for the number of candidate items being always very large. Utilizing the structured properties of the GTAG, we propose an effective and efficient recommendation algorithm named *Breadth-first Preference Propagation* (BPP). The algorithm follows a relaxed breath-first search strategy, and returns recommendation results within 6 propagation steps.

In our experiments on two real-world datasets, the proposed GTAG with BPP significantly outperforms state-of-the-art algorithms including the random-walk based algorithms. To summarize, the main contributions of this paper are three-fold:

- We propose the *Geographical-Temporal influences Aware Graph* for time-aware POI recommendation, which encodes both geographical and temporal information of user check-in records.
- We develop an effective and efficient *Breadth-first Preference Propagation* algorithm for time-aware recommendation on GTAG exploiting both geographical and temporal influences.
- We conduct extensive experiments to evaluate the proposed method on two publicly available LBSN datasets collected from Foursquare and Gowalla, respectively. Experimental results show that our proposed method outperforms state-of-the-art methods for time-aware POI recommendation.

The rest of the paper is organized as follows: we review previous work in Section 2, and report the datasets and the observations on temporal influence in Section 3. In Sections 4 and 5, we introduce the GTAG and the recommendation algorithm BPP, respectively. The experimental results are presented in Section 6. Finally, we conclude this paper in Section 7.

2. RELATED WORK

Collaborative Filtering Collaborative filtering (CF) has been widely adopted for recommender systems. CF exploits users' historical purchase ratings or preference to make recommendations, and it can be divided into two categories, namely, *memory-based* CF and

model-based CF. Memory-based CF methods, such as user-based CF and item-based CF [24], utilize user rating data to calculate the similarity between users or items to make recommendations. In contrast, model-based CF builds recommendation models using data mining techniques, such as matrix factorization [15] and probabilistic topic model [1]. Our proposed method is a graph-based CF method, which falls in the category of model-based. Most of existing graph-based CF methods are based on random walks with restart on the user-item graph [2, 13], while some recent proposals employ different preference propagation strategies on the user-item graph [3, 26].

POI Recommendation Most of POI recommendation studies exploit geographical information. User-based CF and item-based CF are explored in [18, 29, 30]. Ye *et al.* [30] exploit social influence and geographical influence under the framework of user-based CF to make POI recommendations. The authors also study on improving the efficiency for POI recommendations in their earlier work [29]. Levandoski *et al.* [18] employ item-based CF to make POI recommendations with the consideration of travel penalty, which is proportional to the distance between the target user and a POI.

Several model-based approaches have been proposed. Noulas *et al.* [23] find that random walk based approach significantly outperforms MF, user-based CF and item-based CF for POI recommendation. Cheng *et al.* [5] develop a matrix factorization (MF) based model, which incorporates the geographical information by a Gaussian mixture model (GMM). Kurashima *et al.* [16] propose a topic model, in which a POI is sampled based on the topics and the distance to historical POIs visited by the target user. Wang *et al.* [25] propose algorithms under the framework of personalized Pagerank, where the POIs far away from the visited POIs of the target user are filtered out. Yin *et al.* [31] consider the topic differences between cities, and propose an LDA-based model to recommend POIs for a given user at a given city. Liu *et al.* [19] combines probabilistic model with MF to recommend POIs.

Besides the geographical information, some other information has been exploited, such as POI category (tags) [4, 11, 20, 22] and social information [5, 9, 25, 28, 30]. However, POI categories are not always available, and social information has been shown to contribute little to the recommendation performance [5, 30].

Recommendation with Temporal Information A number of time-aware recommendation techniques have been proposed under the framework of decision-tree [33], matrix-factorization [14, 21, 27], item-based CF [8] or graph [26].

Ding *et al.* [8] assume that the recent ratings of users should play a bigger impact on their current preference than do the older ratings. Based on the assumption, when estimating item similarities for item-based CF, they decay the weight of older ratings.

Xiang *et al.* [26] assume that users' purchase behaviors are influenced by their long-term and short-term preferences. They propose a bipartite Session-based Temporal Graph (STG) to model the two kinds of preferences. The STG contains three types of nodes, *user* node, *item* node and *session* node, where the time range is divided into bins and each user-bin pair becomes a session node. A *user* is connected to an *item* node if the user purchased the item, and the item is also connected to the corresponding *session* node. Based on the STG, they propose an Multi-Source Injected Preference Fusion (MS-IPF) algorithm to propagate the preference from the target *user* nodes (reflecting long-term preference) and corresponding *session* nodes (reflecting short-term preference) to candidate *item* nodes. The preference that a candidate node receives via the shortest path is used as its recommendation score. The MS-IPF algo-

rithm is designed for bipartite graphs, and we do not see how it can be applied to general graphs.

Although both MS-IPF [26] and our method are graph-based methods, they are greatly different in both the graph structure and the recommendation technique, which will be detailed in Section 4.3.

Time-aware POI Recommendation Yuan *et al.* [32] propose a model named UTE-SE to make time-aware recommendations, which incorporates time dimension into user-based CF. To address the sparsity problem, they calculate the check-in similarities between different time slots, and employ various techniques to smooth the check-in matrix. They also explore the geographical influence that users tend to visit POIs that are close to their previously visited POI. The scores calculated based on the temporal influence and geographical influence are linearly combined as the final recommendation score for a candidate POI. Different from [32], our proposed GTAG method is a graph-based method that exploits the geographical and temporal influences in an integrated way. Hu *et al.* [12] propose a topic-model based approach named Spatio-Temporal Topic (STT), which exploits the spatio-temporal aspects of user check-ins for time-aware POI recommendation. In STT, each user has distributions over topics and regions, and each time slot has distributions over topics and POIs. The regions, topics and time together influence users’ checkin activities. However, the authors do not provide details for parameter estimation and tuning, which makes it hard to evaluate its effectiveness in the experiments. Gao *et al.* [10] propose location recommendation framework with temporal effects (LRT) to explore temporal influence for POI recommendations. LRT is an MF-based method that factorizes the user-time-POI check-in matrix to get users’ temporal preference matrix U_t in time t and location characteristics matrix L . Then, the POI preference of the target user is constructed and aggregated based on L and U_t in all time slots. LRT does not exploit geographical influence for recommendation. Note that LRT is not used to recommend POIs for a target time in [10]. In Section 6, we tailor it for time-aware POI recommendation by removing the aggregation step, *i.e.*, the POI preference of the target user in target time t is estimated based on L and U_t only.

3. TEMPORAL INFLUENCE ON USER CHECK-IN BEHAVIOR

Since geographical influence (*i.e.*, users tend to visit nearby places, and the willingness of visiting a place decreases with the increase of distance) has been verified in [32], in this section we concentrate on the temporal influence to users’ check-in behaviors on the datasets released by authors of [32].

3.1 Dataset

The two datasets contain check-in records from *Foursquare* and *Gowalla*, respectively. The *Foursquare* dataset contains 342,850 check-ins made in Singapore between Aug. 2010 and Jul. 2011. The *Gowalla* dataset, originally from [7], contains 736,148 Gowalla check-ins made within California and Nevada between Feb. 2009 and Oct. 2010. Each check-in is a $\langle user, POI, time \rangle$ tuple and each POI has its own geographical coordinates.

For a fair comparison with the methods reported in [32], the same data preprocessing is carried on the datasets. That is, the users who checked in fewer than 5 POIs and the POIs which have fewer than 5 users checked in are removed. After preprocessing, the *Foursquare* dataset contains 194,108 check-ins made by 2,321 users at 5,596 POIs, and the *Gowalla* dataset contains 456,988 check-ins made by 10,162 users at 24,250 POIs (see Table 1). Observe that the two

Table 1: Statistics on the datasets (after pre-processing)

Dataset	# Check-ins	# Users	# POIs
Foursquare	194,108	2,321	5,596
Gowalla	456,988	10,162	24,250

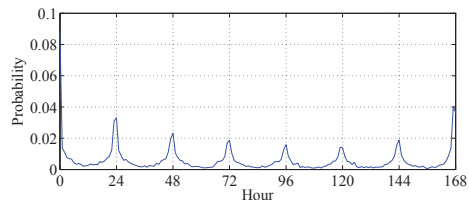


Figure 2: Returning probability on Foursquare dataset

datasets have different scales in terms of the size of entities (*i.e.*, users, POIs, and check-ins) and the geographical range.

3.2 Observation of Temporal Influence

First, we examine the probabilities that users return to their firstly visited POIs after different number of hours. Specifically, if a user visits a POI multiple times, we calculate the time difference between her first check-in at the POI and each of her subsequent check-ins. The time differences of all users at all POIs are aggregated and grouped into bins on hourly basis. Figure 2 plots the probability distribution obtained on the *Foursquare* dataset. Observe that, the returning peaks occur around every 24 hours (*e.g.*, 24, 48, 72, etc.), suggesting that users’ check-in activities are daily periodic. That is, *in close time periods of a day, users are likely to visit the same places, and in different time periods of a day, users visit different places.* This observation is in accordance with the findings reported in [6].

Next, to further understand the periodicity, we discard the date information of the check-ins and compute the deviation hours of each subsequent check-in of a user from the first check-in time of the user at a POI. Figure 3 plots the returning probabilities at different hour deviations on the *Foursquare* dataset. From this probability distribution, it is clear that users tend to visit the same POIs at close time of day, and their behaviors in close time are similar. In addition, the curve fits exponential function well.

Note that, similar observations are made on the *Gowalla* dataset. We choose not to plot the figures due to the page limitation.

4. GEOGRAPHICAL-TEMPORAL INFLUENCES AWARE GRAPH (GTAG)

In this section, we detail the Geographical-Temporal Influences Aware Graph (GTAG) to exploit the observations of geographical influence and temporal influence in time-aware POI recommendation. The notations used in this paper are listed in Table 2.

4.1 GTAG Structure

We build the GTAG based on the following intuitions:

- Intuition 1:** Users’ interests vary with time, and a user may visit different POIs at different time [10, 32]. The temporal interests of a user in a time is reflected as the POIs she visited in that time.
- Intuition 2:** The check-in interests of a user in the time closer to the target time are more relevant, and thus more important, for recommendation.

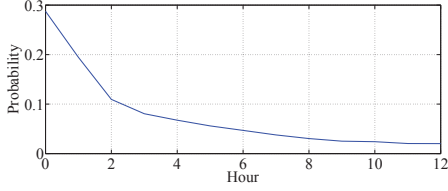


Figure 3: Returning probability on Foursquare dataset (hour deviation)

Table 2: Symbols

Symbol	Description
$G, E, V, P_{i,j}$	GTAG/TAG graph, edge set of G , node set of G , set of paths from node i to node j
$e_{i,j}, w_{i,j}$	edge from node i to node j , weight of $e_{i,j}$
U, L, T, S	user set, POI set, time slot set, session set
$u, \ell, t, s_{u,t}$	user $u \in U$, POI $\ell \in L$, time $t \in T$, session of $u, t \in S$
L_u	visited POI set of u
S_u	session node set of u .
A_n	adjacent node set of n .
$ t_i, t_j $	time difference between t_i and t_j
$dis(\ell_i, \ell_j)$	geographical distance between ℓ_i and ℓ_j
$wi(d)$	the willingness a user visits a d -km far away POI
r_i	preference value of node i
η	the parameter controlling the preference of session nodes propagating to POI nodes and user nodes
τ	the parameter controlling the preference of POI nodes propagating to POI nodes and session nodes
H	the parameter controlling the time influence
k	the maximum number of POI nodes that a POI node can connect to
α, β	the parameters of power law function

3. **Intuition 3:** If two users have similar temporal interests in two time, they tend to visit the same POIs in the two time.

4. **Intuition 4:** Users tend to visit their nearby POIs [30,32].

To represent users' interests at different time (**Intuition 1**), we divide time into 24 hourly slots, and use hour as the time unit. Note that we can adopt different time slots, and our proposed approaches are equally applicable. For example, we can use 48 hourly time slots to differentiate users' activities in weekdays and weekends, where the two sets of 24 hourly slots correspond to the hours in weekdays and the hours in weekends, respectively. The set of time slots is denoted by T . For ease of presentation, *time* and *time slot* are used interchangeably in this paper.

In GTAG, users and POIs are represented by user nodes $u \in U$ and POI nodes $\ell \in L$, respectively. To represent users' check-in interests at different time (**Intuition 1**), we use a session node $s_{i,j} \in S$ to relate the POIs visited by user u_i in time $t_j \in T$. Here U, L and S are the user node set, POI node set, and session node set, respectively. These three types of nodes are connected by weighted directed edges, namely, $E_{U,S}, E_{S,U}, E_{S,L}, E_{L,S}$ and $E_{L,L}$, where $E_{X,Y}$ denotes the set of edges from nodes in set X to nodes in set Y . Edges in the five sets compose two types of links, namely, check-in link and POI link, which embed the temporal and geographical influences, respectively. A check-in link represents a check-in record, and it consists of edges between user node and session node, and between session node and POI node. The edges $e_{u_i, s_{i,j}} \in E_{U,S}$,

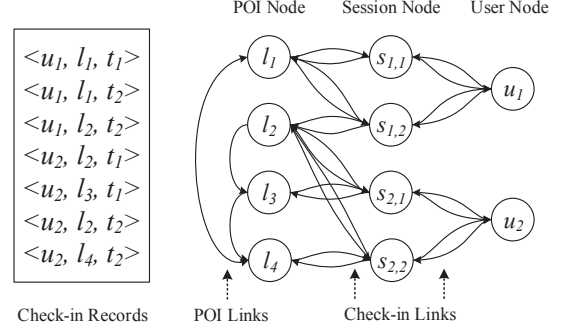


Figure 4: GTAG constructed using the 7 sample check-ins

$e_{s_{i,j}, u_i} \in E_{S,U}$, $e_{s_{i,j}, \ell_k} \in E_{S,L}$, and $e_{\ell_k, s_{i,j}} \in E_{L,S}$ form a check-in link, which corresponds to user u_i 's check-in on POI ℓ_k in time t_j . Since the check-in interests of a user in the time closer to the target time are more relevant (**Intuition 2**), the edges connecting to the session nodes that are close to the target time will be assigned with larger weights (to be explained in Section 4.2). The edges from POI nodes to session nodes bridge sessions of users that share similar POI interests, which enables us to exploit other users' temporal interests for recommendation (**Intuition 3**).

To incorporate the intuition that users tend to visit their nearby POIs (**Intuition 4**), we use a POI link $e_{\ell_k, \ell_m} \in E_{L,L}$ to connect ℓ_k to ℓ_m if ℓ_m is close to ℓ_k in distance. Theoretically, there are $|L||L| - 1|$ edges that link every pair of POIs. Incorporating all of them into GTAG will greatly deteriorate the recommendation efficiency. In addition, if two POIs are far from each other, users are less likely to travel from one to the other, and thus the correlation between them is small. We therefore set a threshold k to limit the number of edges starting from one POI to other POIs, *i.e.*, for each POI, only the k nearest POIs are connected with it.

Figure 4 gives a sample set of 7 check-in records made by two users (u_1 and u_2) on four POIs (ℓ_1 to ℓ_4) during two time slots (t_1 and t_2). The GTAG constructed using this sample set of check-in records is illustrated on the right hand side of the figure. Since u_2 visited POI ℓ_4 in time t_2 (*i.e.*, the last one in the 7 sample check-in records), there are directed edges between u_2 and $s_{2,2}$, and between $s_{2,2}$ and ℓ_4 in the graph. ℓ_3 is connected to ℓ_4 , since ℓ_4 is assumed to be close to ℓ_3 in distance. Note that k is set to 1 in this example.

Generally, GTAG makes recommendations as follows: given a target user u_i and target time t_q , we first set the weights of links from user nodes to session nodes adaptively based on the target time. We inject preference to the user node u_i , and then propagate the preference to candidate POI nodes via various paths. During the propagation, both the geographical and temporal influences are exploited. In the end, the POIs that receive larger preference will be recommended. Before detailing the preference propagation, we first introduce how to set the weights for edges in GTAG.

4.2 Weight Computation of the Edges

The initial weights of all edges in $E_{S,U} \cup E_{S,L} \cup E_{L,S}$ are set to 1. We have also considered setting the weights of edges in $E_{S,L}$ and $E_{L,S}$ based on the number of visits. However, poorer recommendation results were obtained in our experiments compared with using 1 as the weights. This is consistent with the result reported for graph-based recommendation in [23].

Next, we detail how to set the initial weights for edges in $E_{U,S}$ and $E_{L,L}$, and how to normalize the weights for GTAG.

Weights of Edges in $E_{U,S}$. Recall that our time-aware POI recommendation task is to recommend POIs for a target user u to visit at a target time t_q . Naturally, the session node whose time is close to the target time t_q is more important to the recommendation task (**Intuition 2**). Following previous work [8], we use an exponential function to model the importance of the session node for time slot t to the target time t_q :

$$f(t, t_q) = \exp\left(-\frac{1}{H} \cdot |t, t_q|\right), \quad (1)$$

where $|t, t_q|$ is the time difference between t_q and t , and H is a parameter controlling the extent of temporal influence. A smaller H leads to smaller weight for the sessions far from t_q . Note that, if the time slot of session node t matches t_q , then $f(t, t_q) = 1$.

We use the importance values of session nodes computed above to initialize the weights of edges from user nodes to session nodes. Specifically, given a target time t_q for recommendation, the weight of the edge from user u_i to session node $s_{i,j}$ is computed by

$$w_{u_i, s_{i,j}} = f(t_j, t_q), \quad (2)$$

where t_j is the time slot of session node $s_{i,j}$.

We argue that the above weighting scheme offers at least two advantages. First, the weight between a user node and a session node is *adaptively* set based on the target time for recommendation. That is, the GTAG adjusts the importance of session nodes based on the target time for recommendation, so that the temporal influence is considered. Second, the above weighting scheme alleviates the data sparsity problem: when we recommend POIs for target time t_q , check-in records in all time sessions are considered with different weights. If we only consider the check-ins during the target time slot, the data will become much more sparse, and it is well known that sparsity is a major challenge in recommendation.

Weights of Edges in $E_{L,L}$. The observation on graphical influence states that users tend to visit nearby places (**Intuition 4**). The willingness of visiting a place decays with the increase of distance from the current location. Here, we adopt a power-law function of distance to model the willingness of a user moving from one place to another as in [32]. More specifically, the willingness of a user to visit a d -km far away POI is defined by Equation 3:

$$wi(d) = \alpha \cdot d^\beta, \quad (3)$$

where α and β are parameters of the power law function, which can be learned by maximum likelihood estimation.

We employ the willingness as the weight of edge in $E_{L,L}$:

$$w_{\ell_i, \ell_j} = wi(\text{dist}(\ell_i, \ell_j)), \quad (4)$$

where $\text{dist}(\ell_i, \ell_j)$ is the geographical distance between POIs ℓ_i, ℓ_j .

Edge Weight Normalization. After setting the initial weights for edges in GTAG, we normalize the edges' weights as follows:

$$w_{i,j} = \begin{cases} \frac{1}{\tau \sum_{k \in A_i \cap L} wi(\text{dis}(i, k)) + |A_i \cap S|} & \text{if } i \in L \text{ and } j \in S, \\ \frac{\tau wi(\text{dis}(i, j))}{\tau \sum_{k \in A_i \cap L} wi(\text{dis}(i, k)) + |A_i \cap S|} & \text{if } i \in L \text{ and } j \in L, \\ \frac{1}{\eta |A_i \cap L| + 1} & \text{if } i \in S \text{ and } j \in U, \\ \frac{\eta}{\eta |A_i \cap L| + 1} & \text{if } i \in S \text{ and } j \in L, \\ \frac{w_{i,j}}{\sum_{k \in A_i} w_{i,k}} & \text{if } i \in U \text{ and } j \in S. \end{cases}$$

In these equations, A_i is the set of adjacent nodes of node i , $|A_i \cap S|$ is the number of adjacent session nodes of node i , and $|A_i \cap L|$ is the number of adjacent POI nodes of node i . τ is a parameter to balance the propagation preference of a location node to its adjacent location nodes and session nodes. A larger τ indicates that geographical distance plays a more important role in preference propagation. η is another parameter that balances the importance of POI nodes and user nodes to the preference propagation of a session node.

4.3 Comparison with STG

Our proposed GTAG is significantly different from the Session-based Temporal Graph (STG) [26] in at least 4 aspects:

- The item nodes in STG are connected to both user nodes and session nodes, while in GTAG the POI nodes are connected to session nodes and POI nodes. The edges between POI nodes in GTAG enable us to incorporate geographical information for recommendation.
- The session nodes in GTAG are bridged by user nodes while in STG they are not. The fundamental difference lies in the different intuitions in building the graphs. In STG, a user's sessions are independent. If the target user has never checked-in at the target time, the method will fail to make recommendations for the target time. In GTAG, check-ins from all sessions of a user are considered with different weights.
- The temporal preference considered in STG is either long-term or short-term, while in GTAG the temporal preference is periodic (*i.e.*, each user has 24 session nodes).
- STG is a bipartite graph, while GTAG is not. Thus the recommendation method MS-IPF for STG cannot be applied to GTAG.

5. BREADTH-FIRST PREFERENCE PROPAGATION

The basic idea of the preference propagation is to first inject initial preference on the target user node u , and then propagate the preference to candidates POI nodes through various paths over the graph [26]. Defined in [26], the preference propagated by each path p is the production of the initial preference r_u assigned to target user node u and the weights of all edges on the path:

$$r_p^{(p)} = \prod_{e_{i,j} \in p} w_{i,j} \cdot r_u, \quad (5)$$

where $w_{i,j}$ is the weight of the edge $e_{i,j}$ contained in path p . For each candidate POI ℓ , its preference value is the sum of all preference propagated to it through all paths from target user node u :

$$r_\ell = \sum_{p \in P_{u,\ell}} r_p^{(p)}, \quad (6)$$

where $P_{u,\ell}$ is the set of paths from u to ℓ . The top-ranked POIs sorted by preference value are then recommended.

Two *key elements* need to be considered for effective and efficient preference propagation: (i) the selection of the paths among all possible paths from the target user node to POI nodes, and (ii) the algorithm for efficient preference propagation along the selected paths.

In this section, we first present the constraints in selecting paths for preference propagation, and then present an efficient algorithm named *Breadth-first Preference Propagation* (BPP) for preference

propagation. Finally, we analyze the time complexity of BPP in Section 5.3.

5.1 Path Selection

There exist many possible paths between a user node and a POI node. Enumerating all possible paths is computationally expensive, and may introduce noise that will deteriorate the recommendation accuracy [3, 26]. How to select a subset of paths from all possible ones for preference propagation is a key challenge to efficient and accurate recommendation. In our design, we select paths based on the following three criteria:

1. The path must be a simple path, *i.e.*, there is no repeated node in a path. This constraint eliminates loop(s) in a path. For example, in Figure 4, $u_1 \rightarrow s_{1,1} \rightarrow \ell_2 \rightarrow s_{1,2} \rightarrow u_1 \rightarrow \dots$ is forbidden (for looping back to u_1).
2. The path can contain only one visited POI node and session node of the target user. This constraint avoids generating very long propagation paths. For example, in Figure 4, $u_1 \rightarrow s_{1,1} \rightarrow \ell_1 \rightarrow s_{1,2} \rightarrow \ell_2$ is forbidden.
3. The path terminates when an unvisited POI node is met. Without this constraint, the preference will be propagated from an unvisited POI to another unvisited POI, which will amplify the uncertainty in recommendation. In fact, this constraint is in accordance to the user-based CF method, in that only the items purchased by the users who share at least one item with the target user are considered as the candidate set.

The above three criteria determine that a valid path must be of length 3, 4 or 6, as shown in Figures 5(a), 5(b) and 5(c), respectively. Specifically, a preference propagation always starts from a target user node (*e.g.*, u_1), and then visits one of the user's session nodes (*e.g.*, $s_{1,2}$), followed by a visited POI node (*e.g.*, ℓ_2). After that, the preference can be directly propagated to an unvisited POI (*e.g.*, ℓ_3), forming a 3-step path (illustrated in Figure 5(a)). On the other hand, after reaching a visited POI node (*e.g.*, ℓ_2), the next node to visit in a path could be a session node of another user (*e.g.*, $s_{2,1}$, illustrated in Figures 5(b) and 5(c)). At this point, there are two types of propagations: one is to reach an unvisited POI node (*e.g.*, ℓ_3) and stops, which forms a 4-step path as illustrated in Figure 5(b); the other one is to visit the user node of that session node (*e.g.*, u_2), then to distribute the preference to the user's other session node (*e.g.*, $s_{2,2}$), and to reach an unvisited POI node (*e.g.*, ℓ_4). This propagation follows a 6-step path (see Figure 5(c)).

In fact, the 3-step path exploits the geographical influence: the target user u is likely to check-in POIs that are close to her visited POIs (**Intuition 4**). The 4-step path considers the temporal interests of other users: if the set of visited POIs of user u in time t_1 is similar with that of user v in time t_2 , then in t_1 , u is likely to be interested in the POIs visited by v in t_2 (**Intuition 3**). Finally, the 6-step path explores the correlations between time slots: if v shares similar temporal interests with u , then v 's temporal interests that are close to t_q are also important for recommendation (**Intuition 2 and 3**).

5.2 Breadth-first Preference Propagation

With all valid paths selected based on the three criteria, a straightforward propagation method is to adopt the depth-first search (DFS) strategy. Specifically, based on DFS we propagate the preference from the target user node through all possible paths that satisfy the three criteria in Section 5.1. The propagation of a path stops when an unvisited POI node is reached. We name this algorithm *Depth-first Preference Propagation* (DPP). However, the DPP algorithm has high time complexity (See the analysis in Section 5.3) because

the same edge may be visited multiple times along different paths. For example, in Figure 5(c), the edge e_{ℓ_1, ℓ_4} will be visited twice by following the paths $u_1 \rightarrow s_{1,1} \rightarrow \ell_1 \rightarrow \ell_4$ and $u_1 \rightarrow s_{1,2} \rightarrow \ell_1 \rightarrow \ell_4$.

Comparing with DFS strategy, the breadth-first search (BFS) strategy is much more efficient for preference propagation. In BFS, a node n first collects preference from all of its precedent nodes, and then propagates the preference to its subsequent nodes of all paths involving n in a batch manner. For example, following BFS, ℓ_1 first aggregates the preference from $s_{1,1}$ and $s_{1,2}$, and then propagates its received preference to ℓ_4 via edge e_{ℓ_1, ℓ_4} . As a result, the edge e_{ℓ_1, ℓ_4} is visited only once, whereas is visited twice in DFS.

However, BFS cannot be directly applied for preference propagation, because it is designed for graph traversal and each node can be visited only once. In contrast, when propagating preference, some nodes need to be visited multiple times. For example, consider the two 6-step paths in Figure 5(c):

- Path p_1 : $u_1 \rightarrow s_{1,2} \rightarrow \ell_2 \rightarrow s_{2,2} \rightarrow u_2 \rightarrow s_{2,1} \rightarrow \ell_3$
- Path p_2 : $u_1 \rightarrow s_{1,2} \rightarrow \ell_2 \rightarrow s_{2,1} \rightarrow u_2 \rightarrow s_{2,2} \rightarrow \ell_4$

If we follow BFS exactly, p_2 is not valid as $s_{2,2}$ has already been visited in p_1 at the third step. As a result, it cannot be visited at the fifth step in p_2 .

To solve this problem, we relax the constraint on the times of visiting a node, and allow a node being visited multiple times. In fact, the multi-times-visiting problem only happens to the session nodes of users who share at least one POI with the target user (*e.g.*, $s_{2,2}$), because these nodes will be visited at different steps in different paths.

However, after allowing a node being visited multiple times, a new problem arises: the preference a node n' receives from a precedent node n will be propagated back to n , which violates the simple path constraint. For example, u_2 in Figure 5(c) receives preference from $s_{2,2}$ and $s_{2,1}$, and propagates the preference to $s_{2,1}$ and $s_{2,2}$, respectively. However, if we employ BFS, a part of the preference from one node (*e.g.*, $s_{2,1}$) will be propagated back to itself, because at u_2 we cannot differentiate the preference from $s_{2,1}$ between that from $s_{2,2}$. We could solve this problem by keeping a table at each node to record the amount of preference the node receives from each of its precedent nodes, but it will increase space cost significantly and incur extra computation. In this paper, we propose a more efficient solution.

Consider a node n that propagates its preference $r_n \cdot w_{n,n'}$ to its adjacent node $n' \in A_n$. After collecting the preference from its adjacent nodes, n' propagates its preference $r_{n'} = \sum_{n'' \in A_{n'}} r_{n''} \cdot w_{n'', n'}$ to its adjacent nodes $A_{n'}$, one of which is n . Then, the amount of preference originally propagated from n to n' needs to be excluded from the preference to be propagated from n' to n , *i.e.*, the amount of preference that n receives from n' should be:

$$\begin{aligned}
 & \left(\sum_{n'' \in A_{n'}} r_{n''} \cdot w_{n'', n'} - r_n \cdot w_{n, n'} \right) \cdot w_{n', n} \\
 &= \sum_{n'' \in A_{n'}} r_{n''} \cdot w_{n'', n'} \cdot w_{n', n} - r_n \cdot w_{n, n'} \cdot w_{n', n} \\
 &= r_{n'} \cdot w_{n', n} - r_n \cdot w_{n, n'} \cdot w_{n', n}, \tag{7}
 \end{aligned}$$

where the second part is the portion of preference of n that will be propagated back to itself through n' .

This inspires the following method to address the problem. We pre-subtract the amount of preference from n that will be propagated back to it through its neighbors to be visited (*e.g.*, $r_n \cdot w_{n, n'} \cdot w_{n', n}$) after propagating the preference from it (n) to its neighbors. Then, the relaxed BFS can be applied for preference propagation.

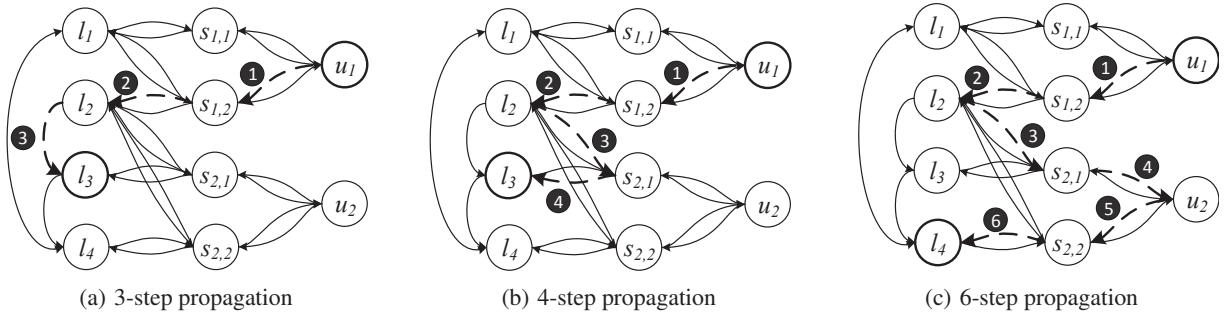


Figure 5: Examples of valid propagation paths of lengths 3, 4, and 6, from a user node u to a location node ℓ . The numbered nodes in black color indicate the propagation steps in each path.

Algorithm 1: Breadth-first Preference Propagation (BPP)

Input: user u , time t , GTAG graph G_t , recommendation size k
Output: Top- k POIs as recommendation results

```

1  $Q \leftarrow$  queue of nodes to be visited;
2  $N \leftarrow$  visited node set;
3 push  $u$  into  $Q$ ;
4 while  $Q$  is not empty do
5    $n \leftarrow$  pop head node of  $Q$ ;
6   if  $n \in N$  then
7     continue;
8   if  $n$  is not a session node of those users who share at least
   one POI with user  $u$  then
9      $n$  add into  $N$ ;
10   $\widehat{r}_n \leftarrow 0$ ;
11  foreach  $n' \in A_n$  do
12    if  $n' \notin \widetilde{L}_u$  and  $n' \notin N$  and  $n' \notin Q$  then
13       $Q.push(n')$ ;
14       $r_{n'} \leftarrow r_{n'} + r_n \cdot w_{n,n'}$ ;
15      if  $n' \in Q$  and  $n \in A_{n'}$  then
16         $\widehat{r}_n \leftarrow \widehat{r}_n - r_n \cdot w_{n,n'} \cdot w_{n',n}$ ;
17   $r_n \leftarrow \widehat{r}_n$ ;
18 return top- $k$  POIs in  $L - L_u$  based on recommendation scores;

```

For example, after performing pre-subtraction for $s_{2,1}$ in Figure 5(c), we can propagate the preference of u_2 to its neighbouring node $s_{2,1}$, since the invalid preference that will be propagated back to it (*i.e.*, $r_{s_{2,1}} \cdot w_{s_{2,1},u_2} \cdot w_{u_2,s_{2,1}}$) has already been subtracted.

We name the proposed propagation approach *Breadth-first Preference Propagation* (BPP), for it is designed based on BFS strategy. The algorithm is shown in Algorithm 2. Line 8 relaxes the constraint on the times of visiting a node. If a node n' is not in the set of candidate POI nodes (\widetilde{L}_u), and needs to be visited in subsequent steps (not in N), we put it into the queue Q (lines 12-13), which contains the nodes to be visited. Line 14 propagates the preference from n to each of its neighbor nodes n' , and lines 15-16 perform the pre-subtraction operation, *i.e.*, if n 's neighbor n' will be visited, and n' preference will be propagated back to n ($n \in A_{n'}$), we subtract this part of preference from n 's preference.

5.3 Time Complexity Analysis

Time Complexity of DPP. We first look at the 6-step path for DPP. In the first step, one of target user u 's session nodes is selected, which has $|T|$ choices. Then, for each session node, we select one

POI node ℓ , which has $|L^{(s)}|$ choices. Here $|L^{(s)}|$ is defined as the maximum number of POIs a session node connects to. In the third step, a session node of other users' connection with ℓ is reached, and there are $|S^{(\ell)}|$ possibilities, where $|S^{(\ell)}|$ is the maximum number of session nodes having links with a POI node. Then, the user node is visited with the cost 1, followed by other session nodes ($|T|-1$). Finally, a POI node is visited from the session node, where there are $|L^{(s)}|$ choices. Based on the product rule, the complexity of DPP is $O(|T||L^{(s)}||S^{(\ell)}||T||L^{(s)}|)$. In the worst case where $|L^{(s)}| = |L|$, $|S^{(\ell)}| = |U||T|$, the complexity becomes $O(|T|^3|L|^2|U|)$. Obviously, the complexities of 3 and 4-step propagations are much lower.

Time Complexity of BPP. For the BPP method, the propagation starts from the target user node u . In the first step, all u 's session nodes are visited, so the complexity is $|T|$. The second steps propagates the preference from each u 's session node $s_{u,t}$ to the POI nodes it connects with by the cost of $|T||L^{(s)}|$. Then, the preference is transformed from u 's visited POI nodes to other users' session nodes they connect with. The cost of the third step is $|L^{(u)}||S^{(\ell)}|$, where $L^{(u)}$ is the maximum number of POIs a user has visited. We define $S_{\ell}^{(\ell)}$ as the set of the session nodes having link with POI ℓ , and $|N_u|$ as the number of users who co-visited at least one POI with u . Then the cost of the fourth step (propagating from other users' session nodes to their corresponding user nodes) and the fifth step (propagating from the other user nodes to their session nodes) will be $|\bigcup_{\ell \in L_u} S_{\ell}^{(\ell)}|$ and $|N_u||T|$. Finally, the preference reaches POI nodes with the cost of $|N_u||T||L^{(s)}|$. For BPP, the total complexity of 6-step propagation is the sum of complexity of each step, which is $O(|T| + |T||L^{(s)}| + |L^{(u)}||S^{(\ell)}| + |\bigcup_{\ell \in L_u} S_{\ell}^{(\ell)}| + |N_u||T| + |N_u||T||L^{(s)}|)$. In the worst case ($|L^{(s)}| = |L|$, $|S^{(\ell)}| = |N_u| = |U|$, $|\bigcup_{\ell \in L_u} S_{\ell}^{(\ell)}| = |U||T|$), the complexity is $O(|T||U||L|)$, which is much lower than that of DPP.

Time Complexity of MS-IPF. The worst case time complexity of MS-IPF method proposed in [26] is $O(|T||U||L|^2)$, which is higher than that of BPP.

6. EXPERIMENTS

We systematically evaluate the proposed method and compare our method with the state-of-the-art methods on 2 real-world datasets. We first detail the experimental setting, then compare the performance of these methods, and finally study the effect of the length of time slot.

6.1 Experimental Setup

Dataset. We conducted our experiments on the two datasets, namely *Foursquare* and *Gowalla* (see Table 1), introduced in Section 3.1, which are the set of datasets used in the work [32].

Table 3: The 11 methods evaluated in our experiments. The \checkmark mark under T or G indicates the method utilizes temporal and geographical influences.

Method	Description	T	G
U	User-based CF (Section 2)	-	-
G-BPP	BPP on TAG without temporal influence	-	-
UG	U with geographical influence [30]	-	\checkmark
UTF	U with time function [8]	\checkmark	-
UTE	Smoothing enhanced time-aware CF [32]	\checkmark	-
UTE-SE	UTE with geographical influence [32]	\checkmark	\checkmark
MS-IPF	Multi Source IPF on STG [26]	\checkmark	-
LRT	Location rec. with temporal effects [10]	\checkmark	-
TAG-RWR	Random Walk with Restart on TAG	\checkmark	-
TAG-BPP	BPP on TAG.	\checkmark	-
GTAG-BPP	BPP on GTAG (Section 4)	\checkmark	\checkmark

By following the setting in previous work [32], we randomly mark off 10% of each user’s visited POIs as *development set* to tune parameters, and mark off another 20% POIs as *test set* to evaluate the effectiveness of the recommendation methods. The remaining 70% POIs form the *training set*. The densities of the training sets of the Foursquare and Gowalla datasets are 6.35×10^{-3} and 9.85×10^{-4} , respectively. As expected, after splitting a day into 24 hourly slots, the data becomes much sparser: 2.65×10^{-4} and 4.10×10^{-5} , respectively.

Evaluation Metrics. To evaluate the accuracy of all methods (*i.e.*, how well a method recovers the hold-off POIs in the test set for a given user at a given time), we use two metrics, namely, Precision@ N and Recall@ N (denoted by Pre@ N and Rec@ N , respectively), where N is the number of top-ranked recommendations. The Pre@ N (Rec@ N) is estimated by averaging the precision(recall) values of all time slots, as it is in [32]. For each metric, we consider three values of N (*i.e.*, 5, 10, 20), where 5 is the default value.

Because both the Foursquare and Gowalla datasets have very low density, it is common that recommendation methods deliver relatively low precision and recall values [13, 30]. In addition, the POIs in the test set of each user may represent only a small portion of POIs that the user may be interested in. Thus, although the low precision and recall values are reasonable, in this paper, we focus on the *relative improvements* we achieved over baseline methods, instead of the absolute values.

Recommendation Methods. We evaluated 11 methods as listed in Table 3. The \checkmark mark in the table indicates a method utilizes temporal and/or geographical influences.

U is the basic CF method without utilizing temporal or geographical influence. We use it to evaluate the improvement brought by considering the temporal/geographical influence. UG is the method proposed in [30], which linearly combines the results of user-based CF and bayesian geographical model to make recommendation. This method does not utilize temporal influence. UTF (user-based CF with time function) is the user-based version of the algorithm proposed in [8]. It estimates the similarity between users as the conventional user-based CF does, but weights the check-ins according to the gaps between their time slots and the target time slot by an exponential time function, which is also used in our method. We also tried the original item-based method in [8], but very poor results were obtained; hence we choose not to report the results. UTE is the time-aware user-based CF proposed in [32]. UTE-SE is the enhanced version version of UTE incorporating geographi-

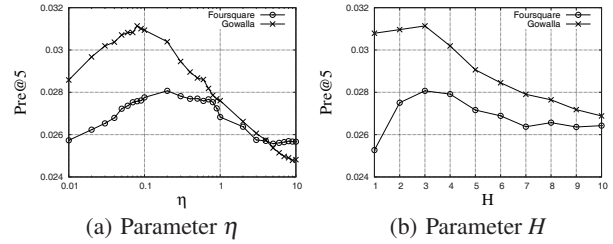


Figure 6: Impact of parameters η and H for TAG-BPP

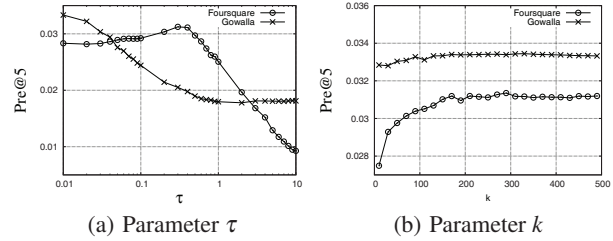


Figure 7: Impact of parameters τ and k for GTAG-BPP

cal influence. MS-IPF [26] and LRT [10] are the recommendation methods using STG graph and MF, detailed in Section 2.

Methods TAG-RWR, TAG-BPP, and GTAG-BPP are all based on techniques proposed in this paper. Specifically, TAG (temporal-aware graph) is a simplified version of GTAG by removing the POI links (*i.e.*, $E_{L,L}$). TAG is employed to fairly compare our preference propagation methods with other methods that do not consider geographical influence. The preference is propagated with Random Walk with Restart (RWR) and BPP algorithms respectively for TAG-RWR and TAG-BPP. To study the performance of our model without temporal influence, we also use a simplified version named G-BPP, in which each user only has 1 (24-hour) session node.

Parameters in all methods are tuned to their optimal values using the development set. For RWR, we set the stop criteria to 0.0001 L-1 distance between vectors of two successive iterations, and the restart probability is 0.15.

6.2 Impact of Parameters

Before comparing with the baseline methods, we tune the parameters of the proposed methods using the development set and examine their impacts. There are four parameters, namely, η , H , τ and k in the proposed GTAG-BPP method. Because TAG is a simplified version of GTAG, we first tune η and H for TAG-BPP on the development set. Then we apply the obtained η and H to GTAG-BPP, and adjust τ and k . The default values of H and k are empirically set to 3 and 500, respectively, when tuning η and τ . Figures 6 and 7 show the Pre@5 of TAG-BPP and GTAG-BPP with varying parameter settings, respectively. Due to the space limitation, we only plot Pre@5. Similar observations hold for Rec@5.

Figure 6 shows that the optimal value of η for Foursquare and Gowalla datasets are 0.2 and 0.08, respectively. Recall that η balances the importance of user nodes and POI nodes when propagating preference from session nodes; a smaller η means more preference will be propagated to user nodes, which will be further distributed to session nodes that are close to the target time. Gowalla dataset is much sparser than Foursquare, and utilizing other session interests can help mitigate the sparsity problem. This might explain why the optimal η for Gowalla data is smaller than that for Foursquare data.

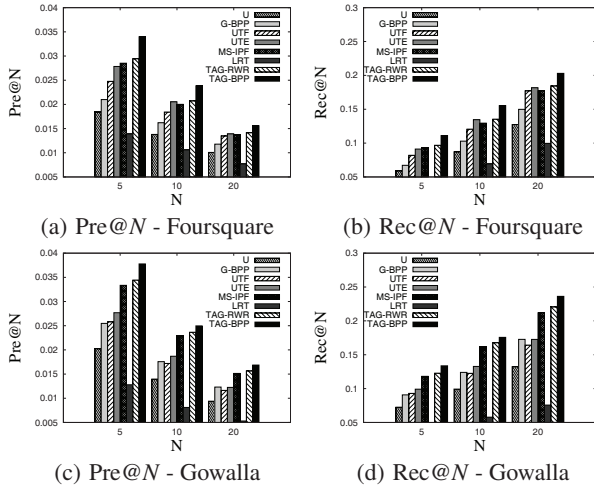


Figure 8: Performance of methods utilizing temporal influence

Parameter H controls the importance of session nodes with respect to the target time. If H is small, the session nodes that are far from the target time will contribute less. The optimal H for both datasets is 3, and accuracy decreases as increasing H , showing that the session interests close to target time are more important. In addition, when H is smaller than 3, poorer accuracy is observed on both datasets, because small H leads to less importance of other session nodes and worsens the sparsity problem.

In GTAG-BPP, parameter τ controls the amount of preference in POI node that will be propagated to its nearby POI nodes. As shown in Figure 7, GTAG-BPP achieves the best accuracy when $\tau = 0.3$ and 0.01 on Foursquare and Gowalla datasets, respectively. To find out the reason for the difference, we compute the average of willingness of visiting from one POI node to its nearby POIs (*i.e.*, the sum weight of POI edges from a POI node). The value is 95.98 and 238.37 respectively on Foursquare and Gowalla datasets. A possible reason is that, in California, users' activities are less sensitive to geographical distance because most people drive cars. Consequently, users are more likely to visit distant POIs. Thus, a smaller τ can weaken the geographical influence. The best k on the two datasets are 290 and 270, which are similar.

6.3 Performance of Methods

We conduct two sets of experiments. The first set of experiments evaluates the accuracy of the methods utilizing temporal influence (U, G-BPP, UTF, UTE, MS-IPF, LRT, TAG-RWR, and TAG-BPP). The second set of experiments evaluate the effectiveness of the methods utilizing geographical influence (UG, UTE-SE and GTAG-BPP).

Methods Utilizing Temporal Influence. The precision and recall of the 8 methods (U, G-BPP, UTF, UTE, MS-IPF, LRT, TAG-RWR, and TAG-BPP) are reported in Figure 8, from which we observe that:

- The proposed method TAG-BPP performs the best w.r.t. both precision and recall at different N values on both datasets.
- Among the memory-based CF methods, UTE achieves the best performance, followed by UTF. U does not exploit temporal influence, and performs the worst. The low accuracy delivered by U suggests that time is an important factor in POI recommendation. Compared against UTE, TAG-BPP

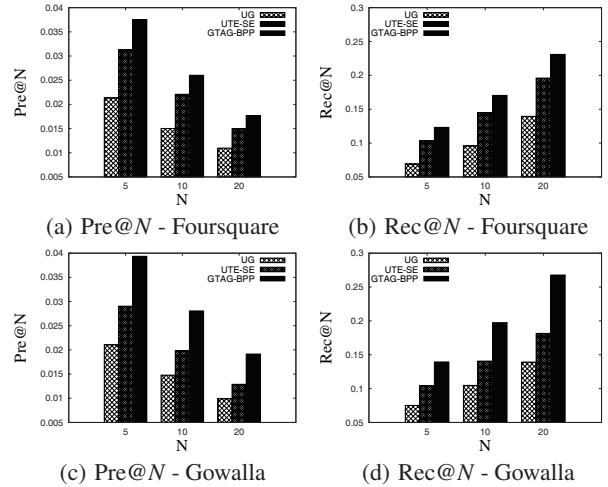


Figure 9: Performance of methods with geographical influence

improves the Pre@5 by 22% and 36% on Foursquare and Gowalla datasets, respectively.

- Among the graph-based CF methods, G-BPP performs the worst because it does not exploit the temporal influence. TAG-RW achieves the best accuracy, probably because of the effectiveness of the TAG graph. Compared against TAG-RW, TAG-BPP improves Pre@5 by 15% and 10% on the two datasets, respectively. In addition, TAG-BPP beats MS-IPF by 19% and 13% w.r.t. Pre@5 on the two datasets. Note that TAG-DPP returns the same results as does TAG-BPP.
- The performance of LRT is not satisfactory, probably because MFmethod does not handle the datasets in low density well. The bad result is also in accordance with that reported in [23].

Methods Utilizing Geographical Influence. The precision and recall of the 3 methods utilizing geographical influence (UG, UTE-SE, and GTAG-BPP) are plotted in Figure 9.

Observe that UG, which exploits geographical but not temporal influence, delivers the worst results. UTE-SE, the state-of-the-art method exploiting both temporal and geographical influences, achieves much better accuracy than UG. GTAG-BPP achieved the best accuracy among the three methods, which outperforms UTE-SE by 20% and 35% in terms of Pre@5 on both datasets, respectively. Compared against TAG-BPP (reported in Figure 8), GTAG-BPP further improves the accuracy by 4% to 10%. The improvement results from the incorporation of geographical influence.

Note that all improvements reported in this sections are significant according to t-test with p -value < 0.01 .

6.4 Effect of the Length of Time Slot

In this section, we study the effect of the length of time slot to recommendation accuracy. When the time slots become longer, the recommendation results will be less time-specific. As in [32], we only consider the methods utilizing temporal influence to focus on the effect of time slot length. We report the Pre@5 and Rec@5 only on the Foursquare data, since the results on the Gowalla data are similar. From Figure 10, we have the following observations.

First, when increasing the length of time slots, all methods achieve better precision, but poorer recall. This is because increasing the length will bring in more groundtruth POIs for each user at each

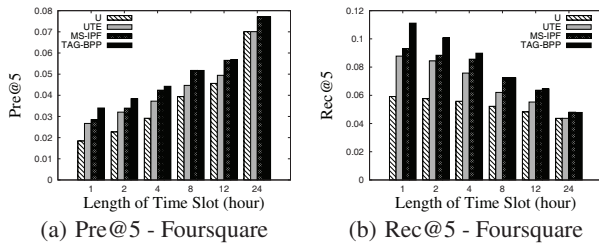


Figure 10: Performance of varying length of time slot

time slot, which makes it easier to capture groundtruth POIs among the top N recommendations, resulting in better precision. However, the number of correctly recommended POIs is constrained by the number of recommendations, which leads to poorer recall values.

Second, for all lengths of time slots, the proposed method TAG-BPP consistently outperforms baselines including the state-of-the-art time-aware recommendation methods MS-IPF and UTE.

7. CONCLUSION

The availability of historical check-in data in LBSNs enables POI recommendation service. In this paper, we focus on the problem of time-aware POI recommendation, which considers the temporal influence in user activities. We propose the GTAG to model the check-in behaviors of users and a graph-based preference propagation algorithm for POI recommendation on the GTAG. The proposed solutions exploit both the geographical and temporal influences in an integrated manner. We conduct extensive experiments over two real-world LBSN datasets. The experimental results show that the proposed methods beat all baselines significantly.

For the future work, it would be interesting to apply the proposed framework to time-aware recommendation in other tasks, e.g., product recommendation on e-commercial websites.

8. ACKNOWLEDGEMENTS

This work is supported in part by a grant awarded by a Singapore MOE AcRF Tier 2 Grant (ARC30/12), a Singapore MOE AcRF Tier 1 Grant (RG66/12), and a grant awarded by Microsoft Research Asia. Quan Yuan would like to acknowledge the Ph.D. grant from the Institute for Media Innovation, Nanyang Technological University, Singapore.

9. REFERENCES

- [1] D. Agarwal and B.-C. Chen. fda: matrix factorization through latent dirichlet allocation. In *WSDM*, pages 91–100, 2010.
- [2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.
- [3] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *WWW*, pages 895–904, 2008.
- [4] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *SIGSPATIAL/GIS*, pages 199–208, 2012.
- [5] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.
- [6] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.
- [7] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, pages 1082–1090, 2011.
- [8] Y. Ding and X. Li. Time weight collaborative filtering. In *CIKM*, pages 485–492, 2005.
- [9] G. Ference, M. Ye, and W.-C. Lee. Location recommendation for out-of-town users in location-based social networks. In *CIKM*, pages 721–726, 2013.
- [10] H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, pages 93–100, 2013.
- [11] B. Hu and M. Ester. Spatial topic modeling in online social media for location recommendation. In *RecSys*, pages 25–32, 2013.
- [12] B. Hu, M. Jamali, and M. Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *ICDM*, pages 1073–1078, 2013.
- [13] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR*, pages 195–202, 2009.
- [14] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.
- [15] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [16] T. Kurashima, T. Iwata, T. Hoshida, N. Takaya, and K. Fujimura. Geo topic model: joint modeling of user’s activity area and interests for location recommendation. In *WSDM*, pages 375–384, 2013.
- [17] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. Slaw: A new mobility model for human walks. In *INFOCOM*, pages 855–863, 2009.
- [18] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *ICDE*, pages 450–461, 2012.
- [19] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *KDD*, pages 1043–1051, 2013.
- [20] B. Liu and H. Xiong. Point-of-interest recommendation in location based social networks with topic and location awareness. In *SDM*, pages 396–404, 2013.
- [21] N. N. Liu, L. He, and M. Zhao. Social temporal collaborative ranking for context aware movie recommendation. *ACM TIST*, 4(1):15, 2013.
- [22] X. Liu, Y. Liu, K. Aberer, and C. Miao. Personalized point-of-interest recommendation by mining users’ preference transition. In *CIKM*, pages 733–738, 2013.
- [23] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *SocialCom/PASSAT*, pages 144–153, 2012.
- [24] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.
- [25] H. Wang, M. Terrovitis, and N. Mamoulis. Location recommendation in location-based social networks using user check-in data. In *SIGSPATIAL/GIS*, pages 364–373, 2013.
- [26] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long- and short-term preference fusion. In *KDD*, pages 723–732, 2010.
- [27] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, pages 211–222, 2010.
- [28] M. Ye, X. Liu, and W.-C. Lee. Exploring social influence for recommendation: a generative model approach. In *SIGIR*, pages 671–680, 2012.
- [29] M. Ye, P. Yin, and W.-C. Lee. Location recommendation for location-based social networks. In *GIS*, pages 458–461, 2010.
- [30] M. Ye, P. Yin, W.-C. Lee, and D. L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.
- [31] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: a location-content-aware recommender system. In *KDD*, pages 221–229, 2013.
- [32] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372, 2013.
- [33] A. Zimdars, D. M. Chickering, and C. Meek. Using temporal data for making recommendations. In *UAI*, pages 580–588, 2001.