2019 Master Thesis

Time Distribution based

Diversified Point of Interest Recommendation

Submission： 2020. 01. 20
Supervisor： Hayato Yamana

Department of Computer Science and Communications Engineering,
Graduate School of Fundamental Science and Engineering,
Waseda University

Student ID：5118F083-1

Fan MO

Abstract

In location-based social networks (LBSNs), personalized point-of-interest (POI) recommendation helps users mine their interests and find new locations conveniently and quickly. It is one of the most important services to improve users' quality of life and travel. Most POI recommendation systems devoted to improve accuracy, however in recent years, diversity of POI recommendations, such as categorical and geographical diversity, receives much attention because a single type of POIs easily causes loss of users' interest. Different from previous diversity related recommendations, in this paper, we focus on visiting time of POI - a unique attribute of the interaction between users and POIs. Users usually have different active visiting time patterns and different frequently visiting POIs depending on time. If a set of proper visiting times of recommended POIs concentrates on a small range of time, the user might be unsatisfied because they cannot cover whole of the user's active time range that results in inappropriateness for the user to visit those POIs. To solve this problem, we propose a new concept— *time diversity* and a time distribution based recommendation method to improve time diversity of recommended POIs. Our experimental result with Gowalla dataset shows our proposed method effectively improves time diversity 25.9% compared with USG with only 7.9% accuracy loss.

Contents

1 Introduction

The exponential increase in data corresponds to the information overload, a problem that leaves users perplexed. Recommendation systems function by filtering information and advising users accordingly. Mobile internet has been developing rapidly, and an increasing number of people use mobile devices. Consequently, numerous location-based social network services (LBSNs) in a wide range of application fields, application fields, such as Yelp[1], Foursquare[2] and TripAdvisor[3], have emerged and attracted millions of users. These services collect users' check-in locations, and allow users to share both the locations and their reviews with friends or the public. Large amount of location data makes it possible to predict and recommend Point-of-Interest (POI) to users. POI recommendation, the main task of LBSN, can benefit users, and has a high commercial value.

Most of the research on the POI recommendation system focuses on improving accuracy. However, emphasis on accuracy and lack of diversity is potentially problematic. Harald Steck [1] states that recommendation systems aiming only at accuracy, capture only the main preference of the user and underrepresent the information of lesser interests, which would gradually narrow down the user's areas of interests. Furthermore, algorithmic confounding [2] occurring in the feedback loop between the user behavior and recommendation will increase homogenization and negatively affect user experience. Thus, diversity is another important metric that needs to be considered in recommendation systems. Specifically, POI recommendation systems emphasizing diversity will provide heterogeneous POIs for the user.

Categorical and geographical influences have been considered in relation to diversity in POI recommendation systems. Proponents of category diversification [3] recommend emphasizing the POIs of different categories to better cover the user's multiple interests. Those of geographical diversification [4], on the other hand, recommend emphasizing the POIs located in a variety of active areas where the user appeared frequently.

Besides the categorical and geographical influence, time is also an important consideration in diversity. In reality, the various POIs not only belong to different categories or geographical areas, they also have different suitable visiting times that is reflected in visiting behavior. Furthermore, the time at which a user visits a POI also

1 https://www.yelp.com
2 https://foursquare.com
3 https://tripadvisor.com

depends on his personal schedule arising from individual peculiarities. For example, a particular user prefers going out in the morning, while another prefers the evening. Besides, the popular time to visit a POI also varies. For example, one POI could be popular at lunch time while another is mainly visited in the night. Considering such time features would increase the time diversity of recommendations, and improve user experience.

In this study, we propose a novel method that focuses on increasing the time diversity of POI recommendations based on the check-in time distributions of the user and POIs. Our proposed method reranks the candidate POI list generated by any existing recommendation algorithms.Our contributions are as follows:

1) Proposal of the concept of *time diversity* and the method of calculating the time distribution vector to represent the preferred time patterns of users and POIs.
2) Proposal of a "time overlap reranking method" which adopts both sliding windows and keeping top-*t* techniques to increase time diversity while keeping accuracy.

In the rest of the paper, related work is introduced in Section 2. The details of our proposed method are presented in Section 3. We show the experiment result in Section 4 and conclude in Section 5.

2 Related Work

Our research focuses on Point of Interest (POI) recommendation and diversity. Thus, in this section, we briefly review existing works on both POI recommendations and techniques for improving recommendation diversity.

2.1 POI Recommendation

Most of the research on POI recommendation focuses on improving recommendation accuracy (precision ratio and recall ratio) by integrating multiple characteristics of the user and POI. The geographical aspect was studied by [5][6][7][8] to describe the user's favorite regions and location of POI. The research [5][9][10] capture the user's preference among different types of POIs based on the categorical aspect. The social aspect was evaluated by [5][11][13], and they conclude that users are more likely to visit the POIs recommended by their friends.

2.2 Diversity of Recommendation

Two aspects of diversity, categorical and geographical, are considered in existing works. The concept of recommendation diversity has been valued by researchers since Ziegler et al. [3] presented categorical diversification to reflect a user's complete spectrum of interests from information retrieval. Vargas et al. [14] proposed a binomial framework to handle category coverage, category redundancy, and the size of the recommendation list. They state that the more categories the recommended items cover, the fewer the redundancies, i.e., no repetition of POIs in the same category. They demonstrated that such diversity improves the users' satisfaction. Based on a novel geographical diversification of the POI recommendation, Han et al. [4] adopted the proportional method to make the number of recommended POIs for each area proportional to the users' activity in it.

Two main techniques, i.e., reranking and diversity-oriented techniques, are used to improve diversity of POI recommendation. The reranking-based technique re-arranges the order of items in a candidate list generated from an existing recommendation algorithm. The advantage of reranking is flexibility, because it considers the base algorithm as a black box that can be replaced freely. The diversity-oriented recommendation designs new algorithms to optimize diversity directly, usually increasing diversity.

The reranking-based techniques [22][23] select recommended items from a large candidate list generated from a base algorithm that is considered a black box function. Usually, greedy selection is adopted to avoid combination explosion and improve the efficiency of the reranking technique. An *n-size* candidate list is scanned $k(k<n)$ iterations, and in each iteration, the item with the highest objective score is selected as part of the user's final recommendation list. Barraza-Urbina et al. [15] proposed an exploitation-exploration diversification method called *XPLODIV* to postfilter the candidate items to select a subset of diversified relevant items. Exploitation reinforces items that can represent the user's previous preference while the exploration allots higher scores to novel items that are different from the user's previous interests to encourage the user to explore unknown. Harald Steck [1] advocated the reranking technique for calibrated recommendation (e.g., if the user watched 70 romance movies and 30 action movies, the recommended movies should be composed of 70% romance movies and 30% action movies) to take into account the user's main interests, without crowding out his lesser interests. However, these previous works did not take time diversity into account; hence, our study is an attempt to fix this gap.

Unlike the reranking-based techniques, the diversity-oriented recommendation [19][20][21] advocates achieving the goal of diversity by designing new algorithms that optimize diversity directly when generating the recommendation lists. The method treats the diversity impact as part of the model. Zhao et al. [16] proposed an algorithm called *CBRS* that integrates the Wundt curve in psychology with Matrix factorization. They indicated that the probability of too familiar or too boring items being chosen by the user was low. Eskandanian et al. [17] claimed that users' preference or tolerance for diversity varies greatly, and the recommendation system therefore ought to capture the different needs for diversity. Based on this assumption, they proposed a method for clustering users based on the level of their desire for diversity, and then the user-based kNN algorithm only considered users in the same cluster as neighbors. Same as Eskandanian's work, Zanitti et al. [18] also adopted a clustering technique as part of their framework. They proposed a novel concept—distant neighbors, which is defined as user's neighbors within the same cluster but far from target user. They claimed that generating recommendation from distant neighbors can improve item diversity. However, similar to the reranking technique, all of them neglect time diversity in their works.

2.3 Time Aspect

Recently, [9][13] proposed improving the POI recommendation accuracy based on the time factor. Yuan et al. [30] first suggested taking the time factor into account to improve POI recommendation accuracy. They suggested splitting a day into several time slots, and extending the user-based collaborative filtering model by integrating the user's activity in each time slot. When calculating the similarity between two users, the individual visit time is considered. Thus, the similarity between two users who visited the same POI becomes low if their visiting times are different. That is, two users who visit the same POI at the same time slots have high similarity.

Yao et al. [12], on the other hand, propose employing the time factor to characterize the attributes of POIs. For each POI, they proposed a method to profile the "temporal area popularity" and "temporal category popularity" in different time slots. The temporal area popularity is calculated from the frequency of the check-ins of all the users at the POIs in an area in each time slot, based on the assumption that users concentrate on different areas at different times throughout a day. The temporal category popularity shows the variation over time in the popularity of different categories of POIs. For example, restaurants are popular at lunch time while bars are popular at night.

However, these research aim to improve the accuracy based on the time aspect, but neglect the importance of time diversity in relation to POI recommendation.

2.4 Summary

In section 2, we introduce existing methods on both POI recommendations and techniques for improving recommendation diversity. Mainly two techniques (reranking-based technique and diversity-oriented recommendation) were adopted to improve diversity. The summary and comparison of these two techniques are shown in Table 1.

Table 1: Summary of Diversity of Recommendation

<table>
<tr><td>Method</td><td>Technique</td><td>Weakness</td></tr>
<tr><td>Harald Steck[1]</td><td>Reranking-based</td><td rowspan="7">1) Compared with Diversity-oriented recommendation, usually have lower diversity
2) Do not take time diversity into account</td></tr>
<tr><td>Ziegler[3]</td><td>Reranking-based</td></tr>
<tr><td>Han et al. [4]</td><td>Reranking-based</td></tr>
<tr><td>Vargas et al. [14]</td><td>Reranking-based</td></tr>
<tr><td>Valcarce et al. [22]</td><td>Reranking-based</td></tr>
<tr><td>Benouaret et al. [23]</td><td>Reranking-based</td></tr>
<tr><td>Barraza-Urbina et al. [15]</td><td>Reranking-based</td></tr>
<tr><td>Zhao et al. [16]</td><td>Diversity-oriented</td><td rowspan="6">1) Compared with Reranking-based technique, need to design a new algorithm
2) Neglect time diversity</td></tr>
<tr><td>Eskandanian et al. [17]</td><td>Diversity-oriented</td></tr>
<tr><td>Zanitti et al. [18]</td><td>Diversity-oriented</td></tr>
<tr><td>Lee et al. [19]</td><td>Diversity-oriented</td></tr>
<tr><td>Cheng et al. [20]</td><td>Diversity-oriented</td></tr>
<tr><td>Wasilewski et al. [21]</td><td>Diversity-oriented</td></tr>
</table>

3 Proposed Method

To improve time diversity of POI recommendation system, we adopt an existing recommendation algorithm to generate a large candidate list that is then used to rerank and select final POIs recommendations. For the reranking strategy, we propose a time overlap-based method, using the diversified time distributions of POIs to cover the user's active periods.

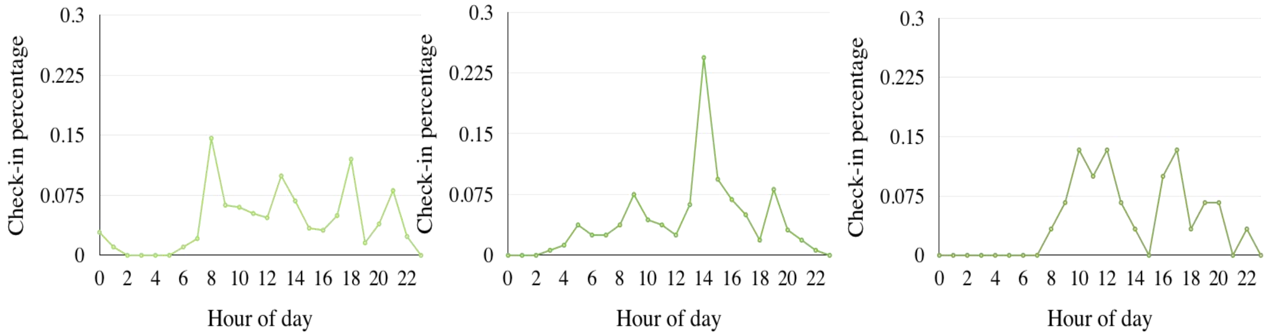
3.1 User's Active Time Distribution

Each user has a distinct activity behavior. The check-in history of users exhibits a unique pattern in the aspect of time. After splitting a day into equal-sized 24 slots, for each user, we use a 24-dimension vector $DUSER_u = \{pu_u^0, pu_u^1, \ldots, pu_u^{23}\}$ to indicate user u's visit behavior in time aspect. pu_u^i means the check-in percentage of user u's visits happening between *i* o'clock and *i+1* o'clock in a whole day, calculated from user u's history (note that 24 o'clock equals 0 o'clock). Here, $\sum_{i=0}^{23}(pu_l^i)$=1 holds. Fig. 1 shows three typical users' active time distributions in real dataset. In Fig.1, *User2* prefers to check in at 14 o'clock while *user3* prefers to visit POIs at 10 o'clock and 12 o'clock. These characteristics can be captured by user's time distribution vector.

3.2 Time Distribution of POI

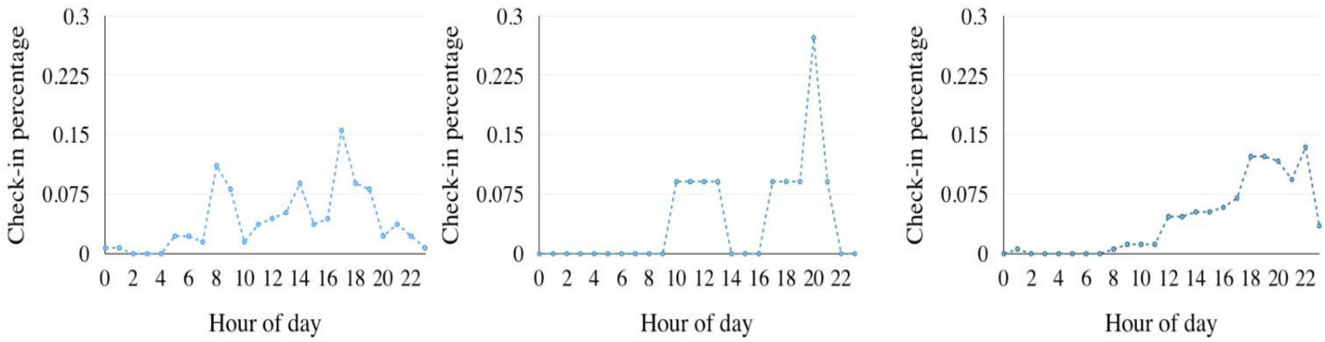
Similar to the user's visit behavior in terms of time, POIs also have different visiting patterns, i.e., the frequency of visits to different types of POIs peaks at different time slots. We also define a 24-dimension time distribution vector $DPOI_l = \{p_l^0, p_l^1, \ldots, p_l^{23}\}$ for POI l to indicate visiting pattern. p_l^i means the percentage of all users' visits happening

between *i* o'clock and *i+1* o'clock, and $\sum_{i=0}^{23}(p_l^i)$=1 holds. Fig. 2 shows the typical visit time distribution of three typical POIs. *POI 1* is most frequently visited at 8 o'clock and 17 o'clock while most visits to *POI 2* are concentrated from 19 o'clock to 20 o'clock. POI 3 is most visited from 18 o'clock to 22 o'clock.



(a) User 1 (b) User 2 (c) User 3

Figure 1: Different users' check-in time distributions

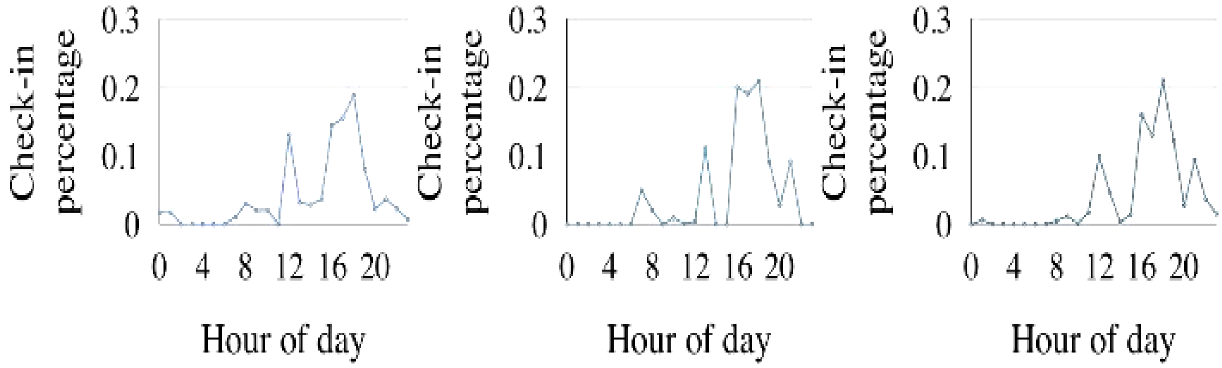


(a) POI 1 (b)POI 2 (c)POI 3

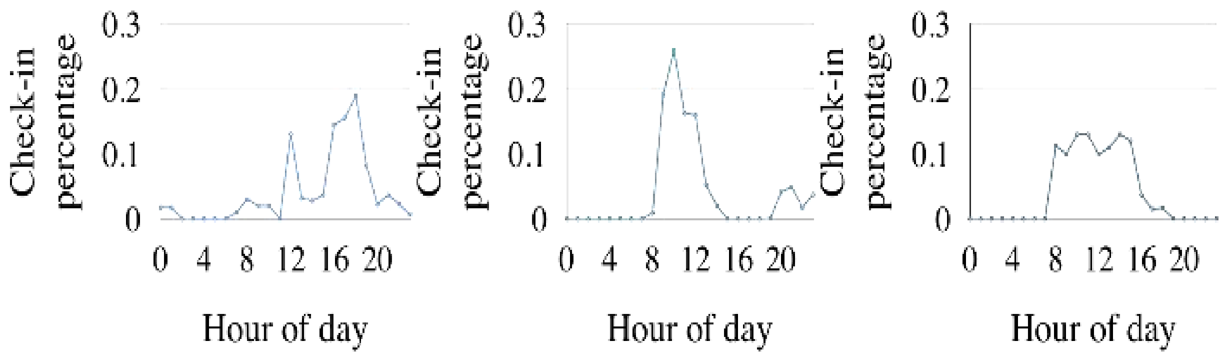
Figure 2: Different POIs' visited time distributions

3.3 Time Diversity

Time diversity is defined as the difference of the time distributions among recommended POIs, measured by the intra-list diversity (ILD)[3]. An example is shown in Fig. 3, the comparison between two sets of POIs to explain time diversity. There are three POIs in each case, and their time distributions are plotted. In Case 1, the time distributions of the POIs are almost the same, with slight difference in their peak time; In Case 2, the frequencies of the check-in time of the three POIs are different, thus the POIs have few intersections between each pair of time distributions. Case 2 has a higher time diversity than Case 1.



Case 1



Case 2

Figure 3: Different POIs' visited time distributions

3.4 Time Overlap Reranking Method

Our research goal is "maximizing time diversity of selected POIs while keeping accuracy." Here, the difference between time distributions is measured by ILD. To achieve the goal, we propose a *time overlap reranking method* that is based on two ideas of ours: 1) when reranking, we will be able to keep the accuracy more by covering the user's time distribution with selected POIs' time distribution; 2) the time diversity among POIs' time distribution will increase when the overlap among the time distribution of the selected POIs is small.

As for the second idea, we have confirmed the relationship between the overlap of the POIs' time distribution and their ILD. We selected 20,000 pairs of POIs from the Gowalla dataset, and calculated the overlap and ILD of each pair. As shown in Fig. 4, the overlap and ILD are negatively correlated with the Pearson coefficient ρ= -0.83. Thus, a smaller overlap corresponds to improved time diversity.

Based on the above ideas, for each user, we select k POIs from his candidate list, which consists of n POIs, generated from a base algorithm. The technique we use is the greedy reranking, selecting k POIs in k selection iterations. In each iteration, one POI with the maximum score is selected for inclusion in the user's final recommendation result list. The score becomes large, 1) when the selected POI's time distribution overlaps the user's time distribution, and 2) the selected POI has small overlaps (intersection area) with the time distribution of the already selected POIs. Although we must calculate the overlap area between each of the already selected POIs and each of the candidate POIs, it is time-consuming; thus, we only calculate the overlap between the accumulated distributions of already selected POIs and each of the candidate POIs.

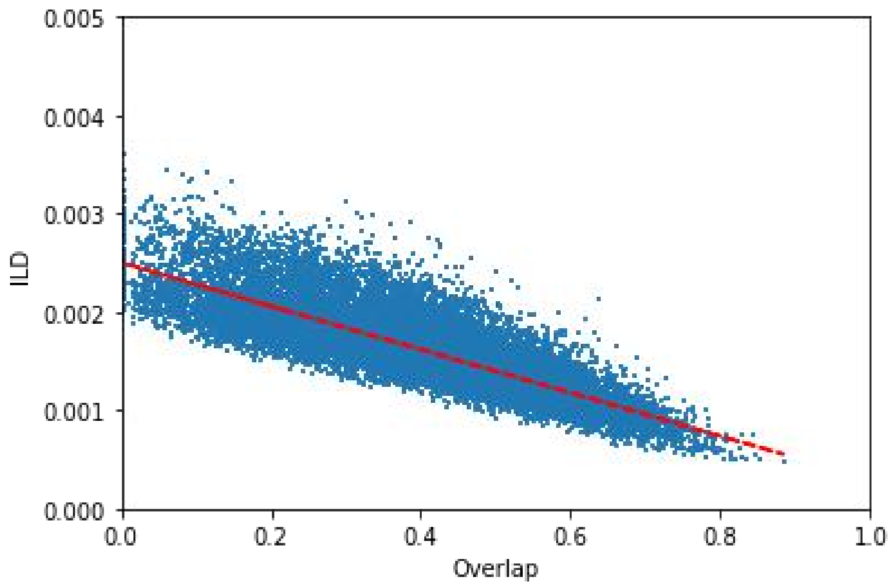


Figure 4: Relationship between Overlap and ILD

The POIs that have been selected in the user u's recommendation list after j iterations(j<k), denoted as $RecList_u^j$, are composed of accumulated distribution $DACC_u^j$. In each time slot, the maximum check-in percentage of the recommended POIs is maintained, to indicate the accumulated value in the slot, as shown in Formulas (1) and (2).

$$DACC_u^j = \{peak_u^0, peak_u^1, \ldots, peak_u^{23}\} \tag{1}$$

$$peak_u^m = \max\,(p_{l_0}^m, p_{l_1}^m, \ldots, p_{l_j}^m) \mid l_x \in RecList_u^j \tag{2}$$

, where $peak_u^m$ is the maximum value of selected POIs' time distribution on time slot m in the user u's recommendation list. $RecList_u^j$ is the user u's recommendation list after j selection iterations. To score each POI in the user's candidate list, we compare $DACC_u^j$ and the user's time distribution. As shown in Fig.5, the overlap areas represent the overlap among time distributions. We define four kinds of area, as shown in Table 2.

To evaluate the quality of each area, we make two assumptions as follows:

1) Larger overlap areas between candidate POI's distribution and user u's time distribution will keep recommendation accuracy, because larger overlap means a higher possibility of the user visiting the POI during his active time.

2) Larger overlap areas between candidate POI's distribution and the accumulated distribution decrease time diversity, because larger overlap area indicates more similarity between the distribution of selected POIs and that of candidate POI, that is the candidate POI is probably redundant.

Table 2: Four kinds of area

Overlap / Area	User's time distribution	Accumulated POIs time distribution
Area1	Yes	No
Area2	No	No
Area3	Yes	Yes
Area4	No	Yes

Based on these two assumptions, we consider area1 is the best, on account of keeping both accuracy and time diversity, followed by area2 and area3, which can maintain accuracy or time diversity respectively. Area4 is the worst area because it can neither keep accuracy nor time diversity.

Given a user u, for each candidate POI l, after calculating the sizes of these four areas, we score the POI using linear combination, as follows:

$$Score_{u,\ l} = \alpha Area1_{u,l} + \beta Area2_{u,l} + \gamma Area3_{u,l} + \delta Area4_{u,l} \tag{3}$$

, where $\alpha, \beta, \gamma, \delta$ are four parameters to balance the importance of four areas. Here we set $\alpha > \beta > \gamma > \delta$.

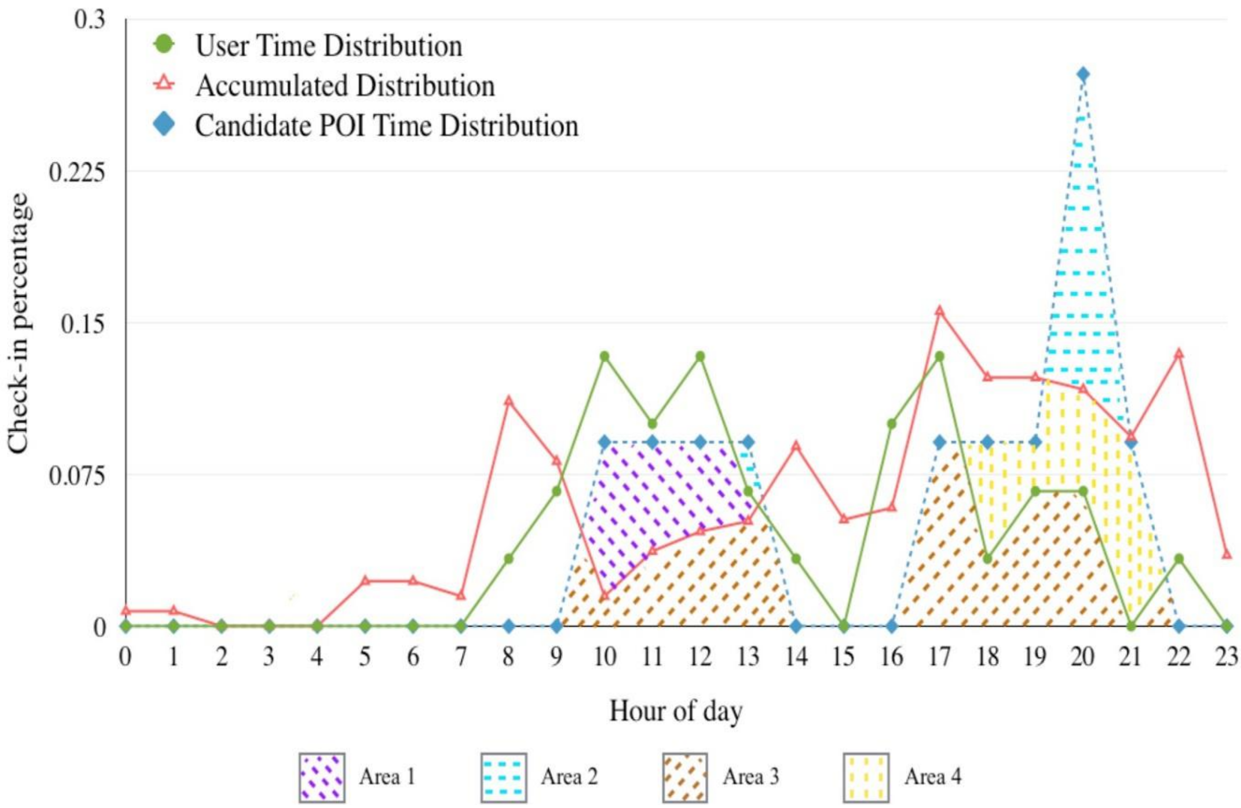


Figure 5: Example of Time Overlap

3.5 Sliding Window

Our proposed method in Sub-section 3.4 can select a set of time diverse POIs from the candidate list. Because the accumulated distribution maintains the maximum value of the time distributions of the recommended POIs in each time slot, the accumulated distribution $DACC_u^j$ becomes larger as more POIs are selected; consequently, most areas of the user's time distribution are covered, i.e., Area 1 is almost nonexistent. Thus, POI selection in our method tends to be more challenging, because the scores are close.

To solve this problem, we propose to use sliding window. With a *w-size* sliding window, the accumulated distribution of last w POIs in recommendation list is kept. When a POI is selected into user's recommendation list, the window slides backward, as shown in Table 3. Sliding window can drop some accumulated distribution, making our proposed method in Sub-section 3.4 more effective to score candidate POIs. We conducted an experiment of selecting 1,000 users from tuning data set, and confirmed their accumulated distribution. In average case, 4 selected POIs cover 90% of user's time distribution. Thus, we set the window size as 3. In this case, sliding window with size 3 starts to work after the forth POI was selected, which means sliding window starts to work after most of user's distribution has been covered. Note that the selected

Table 3: Example of sliding window with size 2

Iteration	Selected POI	Accumulated POI
1	$\boxed{POI_7}$	POI_7
2	$\boxed{POI_7\ POI_3}$	$POI_7\ POI_3$
3	$POI_7\ \boxed{POI_3\ POI_4}$	$POI_3\ POI_4$
4	$POI_7\ POI_3\ \boxed{POI_4\ POI_9}$	$POI_4\ POI_9$

POIs do not be removed from user's recommendation list, we only use sliding window to score candidate POI.

3.6 Keeping Top t

We consider that a base algorithm has higher accuracy for POIs at the top of candidate recommendation list. For POIs at the bottom of the candidate recommendation list, accuracy is somewhat lower than the first few. Thus, we adopt the strategy that retains the top-*t* POIs in the user's candidate recommendation list to further maintain recommendation accuracy. For the other *n-t* POIs (*n* is the size of the candidate recommendation list), we apply the proposed method described in Sub-sections 4 and 5 to select them from the user's candidate list.

3.7 Summary

In section 3, we introduce user's and POI's time distribution, along with three techniques ('Time Overlap Reranking Method', 'Sliding Window', and 'Keeping Top t') to improve time diversity while keeping recommendation accuracy. Algorithm 1, 2 show the pseudo code of 'Time Overlap Reranking Method' and 'Sliding Window'. Algorithm 1 selects a POI with highest score in one iteration using greedy selection. After getting four kinds of overlap areas, the score is calculated as formula (3). Then, accumulated distribution is updated as formula (1), specifically, when we adopt sliding window technique, the updating process of accumulated distribution is shown as algorithm 2.

Algorithm1: Time Overlap Reranking Method

Input: *DUSER*, *Cand_List*, *DPOI, U*

Output: *RecList*

//*U:* a set of users

k: the length of recommendation list

//*DUSER*: users' time distributions

Cand_List: candidate list of all users

//*DPOI*: POIs' time distributions

RecList: recommendation list of all users

//*DACC*: users' accumulated distributions

$Area_i$: four kinds of defined areas

//Calculate Overlap: a function to calculate overlaps between $DPOI_l$ and $DUSER_u$, and between $DPOI_l$ and $DACC_u$

//α, β, γ, δ: four parameters

for each user $u \in U$
    do $DACC_u$ [0,0,0,…,0] //initialize accumulated distribution
        result []
        for j 1 to k //select one POI in each iteration using greedy selection
            do max_score 0
                selected_POI NULL
                for each POI *l* in $Cand\_List_u$ //consider all POIs in candidate list
                    do Area Calculate Overlap($DACC_u$, $DUSER_u$, $DPOI_l$)
                        score $\alpha Area1_{u,l} + \beta Area2_{u,l} + \gamma Area3_{u,l} + \delta Area4_{u,l}$
                        if score > max_score //record the POI with max score
                            then max_score score
                                selected_POI l
                append selected_POI to result
                update $DACC_u$ with *DPOI* selected_POI
                remove selected POI from $CandList_u$
        $RecList_u$ result //save the result
return *RecList*

Algorithm2: Sliding Window

Input: *RecList*, j, w, *DPOI*

Output: updated $DACC_u$

//*RecList*: recommendation list of all users   w: the size of sliding window

//j: the size of already selected POIs $DACC_u$: user u's accumulated diatribution

//$DPOI_l$: POI l' time distribution

for i   1 to 24

    do $peak = \max(\{DPOI_l[i] \mid l \in RecList_u[\max(0, j - w) .. j]\})$

        $DACC_u$[i]   *peak*   //update accumulated distribution

return $DACC_u$

4 Preliminary Evaluation

4.1 Dataset and Base Algorithms

To evaluate the performance of the proposed method, we chose Gowalla[4] as dataset and two base algorithms: USG [6] and LFBCA [24]. Gowalla is a large-scale public dataset of POI check-ins. The main difference between Gowalla and other frequently used datasets such as Foursquare[5] and Yelp[6] is that all check-ins of Gowalla dataset include the timestamp when the check-in happens, by which we can exploit time distribution information to improve diversity.

We directly adopt both the data preprocessing and the base algorithms, USG and LFBCA, implemented by Liu et al. [25] to generate the candidate list. We then apply our proposed reranking method to the candidate list to generate a new recommendation result list that is then evaluated.

Following the convention, the users and POIs with less than 10 check-ins are filtered out. After preprocessing, the number of users is reduced from 25,379 to 18,737, and the number of POIs is reduced from 32,623 to 32,510, respectively. We also follow the pre-defined partition of training, tuning and testing set by Liu et al. [25]. For each user, earliest 70% of check-ins are used as training set, latest 20% of check-ins as testing set and rest 10% as tuning set.

We pre-evaluated the performance of several original base algorithms in term of accuracy, including LORE [11], LRT [26], MGMPFM [27], USG, and LFBCA. The USG and LFBCA outperformed other base algorithms; thus, we chose both. USG is a hybrid method considering user preference, social link, and geographical influence. Although USG was proposed in the early stage of research on the POI recommendation, the performance is still good enough. The LFBCA, another base algorithm, is a link-based method that models user preference and social link in a graph. It achieves similar performance with USG on the Gowalla dataset.

The base algorithms are trained using the training set, and used to generate the candidate recommendation list with *n* POIs. We rerank the candidate POIs, as described in Section 3, and select *k* POIs as the recommendation result. In the experiment, we

4 http://snap.stanford.edu/data/loc-gowalla.html
5 https://sites.google.com/site/yangdingqi/home/foursquare-dataset
6 https://www.yelp.com/dataset_challenge

assume n=20 and k=10. For each user, we acquire the candidate lists generated by the base algorithm, rerank the candidate POIs, and evaluate the recommendation result. The parameters, $\alpha, \beta, \gamma,$ and δ, are searched for in the tuning set using the Bayesian optimization. We chose $\alpha = 0.82, \beta = 0.57, \gamma = 0.34,$ and $\delta = 0.08$ for the USG; and $\alpha = 0.71, \beta = 0.52, \gamma = 0.50,$ and $\delta = 0.01$ for the LFBCA.

In addition, we also evaluate the method that randomly reranks the candidate list for comparison. To observe the effect of keeping top t technique proposed in Section 3, we also evaluate partly random method that keeps top t POIs in candidate list, and randomly selects other POIs as result.

4.2 Evaluation Metrics

We choose 3 metrics to evaluate our method: *Prec@K*, *ILD@K* and *Serendipity@K*. We calculate each metric for each user's final recommendation result, candidate recommendation list and ground truth and then compare the average value of each metric for all the test user.

Prec@K is a widely used metric for evaluating accuracy; it is defined as Formula (4), where GT is the ground truth, representing the places that the user actually visited, and $RecList_K$ is the recommendation list with length K generated by our proposed method.

$$Prec@K = \frac{|RecList_K \cap GT|}{|RecList_K|} \tag{4}$$

ILD@K [3] measures the pairwise dissimilarity and indicates diversity, calculated as formula (5). In our context of time diversity, the dissimilarity is the Euclidean distance between time distribution vectors of POIs that can also be expressed as L^2 norm as formula (6).

$$ILD@K = \frac{\sum_{i,j \in RecList_K \wedge i \neq j} dissim(i,j)}{|RecList_K| * (|RecList_K| - 1)} \tag{5}$$

$$dissim(i,j) = \left\|DPOI_i - DPOI_j\right\|_2 = \sqrt{\sum_{h=0}^{23}(p_i^h - p_j^h)^2} \tag{6}$$

Serendipity@K [28] is the portion comprised of both checked and unexpected POIs in the users' recommendation list, as shown in Formula (7). If the POI appears in the ground truth, that means user has checked it. Intuitively, the POI is unexpected if its frequently visited time is different from the user's active time. Thus, we calculate the average distance between the distribution of users and all candidate POIs; then, we classify the POIs with an above average difference in its time distribution and that of the user as unexpected. We define the unexpected POI in Formula (8) as follows. First, $AvgDis_u$, the average Euclidean distance between the time distribution of the user and time distribution of POIs, in the candidate list is calculated as shown in Formula (9); then, for the POI in the recommendation list, the distance between the time distribution of the POI and user is calculated, as shown in Formula (10). Finally, if the distance is larger than $AvgDis_u$, the POI is considered as unexpected. The recommendation of an unexpected POI that is useful is serendipitous, which impacts the user experience of the recommendation system.

$$Serendipity@K = \frac{|GT \cap RecList_K \cap UNEXP_u|}{|RecList_K|} \tag{7}$$

$$UNEXP_u = \{l | distance(DPOI_l, DUSER_u) > AvgDis_u\} \tag{8}$$

$$AvgDis_u = \frac{\sum_{l \in CandList_u} distance(DPOI_l, DUSER_u)}{|CandList_u|} \tag{9}$$

$$distance(i,j) = \left\|DPOI_i - DUSER_j\right\|_2 = \sqrt{\sum_{h=0}^{23}(p_i^h - pu_j^h)^2} \tag{10}$$

4.3 Results and Discussion

We conclude our experiment result in Table 4. To make results easier to compare, Fig.

6, Fig.7, Fig.8 show the experiment result of Precision@10, ILD@10, and Serendicipy@10 respectively with USG base algorithm while Fig. 9, Fig.10, Fig.11 show the experiment result of Precision@10, ILD@10, and Serendicipy@10 respectively with IFBCA base algorithm. We evaluate the final reranked recommendation list with length 10, which is generated from base algorithm with original length of 20 (i.e., n=20 and k=10). Different configurations of using sliding windows and keeping top t technique mentioned in Section 3 are also compared. We test 4 configurations of enabling and disabling sliding windows with size 3 and keeping top 3. The maximum value for each metric is noted bold. For both base algorithms, our proposed method achieves higher *ILD@K* and *Serendipity@K*. Although the precision of our proposed method is lower than original base algorithm due to the nature of reranking, the decrease level is acceptable, with similar degradation as [29] (<10%, comparing proposed keeping top3 and sliding window 3 with base algorithm). In other words, we traded off a little bit (-0.0038, -7.9%) precision for significant improvement (+0.0465, +25.9%) in diversity with USG. With LFBCA, we also traded off a little (-0.0042, -9.2%) precision for significant improvement (+0.0440, +17.9%) of diversity.

Note that the ILD and serendipity of our method outperforms not only the original baseline but also the randomized version, which indicates our time-overlap reranking algorithm can improve diversity.

To address the effect of the sliding window and keeping-the-top-t technique, we can compare different configurations with and without those techniques: methods deploying the sliding window achieve higher ILD, compared with those not deploying it; therefore, the sliding window technique can significantly increase the diversity during the reranking process. The methods that kept the top t achieve higher precision than corresponding methods that do not keep the top t; thus, keeping top t technique can reduce the loss of precision. Therefore, the configuration where the sliding window is enabled and the top t is kept is a tradeoff between diversity and accuracy, which should depend on the scenario and attention of application.

4.4 Summary

In section 4, we choose 2 base algorithms (USG and LFBCA) and evaluate our proposed method on Gowalla dataset, including 18,737 users and 32,510 POIs after preprocessing. Three metrics (*Prec@K*, *ILD@K*, *Serendipity@K*) are chosen to evaluate the performance of the proposed method. Experiment result shows our proposed method effectively improves (+25.9% using USG, +17.9% using LFBCA) time diversity with

Table 4: Evaluation result

Base Algorithm	Reranking Configuration		*Prec@10*		*ILD@10*		*Serendicipy@10*	
USG[6]	Baseline	(original USG) without reranking	**0.0482**		0.1796		0.0153	
		random	0.0381	(-0.0101)	0.1963	(+0.0167)	0.0150	(-0.0003)
		random with keeping top 3	0.0434	(-0.0048)	0.1901	(+0.0105)	0.0145	(-0.0008)
	Proposed time overlap reranking method	without sliding window or keeping top	0.0398*	(-0.0084)	0.2189*	(+0.0393)	0.0183*	(+0.0030)
		with keeping top3	0.0449*$^{+}$	(-0.0033)	0.2188*$^{+}$	(+0.0392)	0.0175*$^{+}$	(+0.0022)
		with sliding window 3	0.0394*$^{+}$	(-0.0088)	**0.2372***$^{+}$	**(+0.0576)**	**0.0201***$^{+}$	**(+0.0048)**
		with keeping top 3 and sliding window 3	0.0444*$^{+}$	(-0.0038)	0.2261*$^{+}$	(+0.0465)	0.0181*$^{+}$	(+0.0028)
LFBCA [24]	Baseline	(original LFBCA) without reranking	**0.0459**		0.2461		0.0146	
		random	0.0366	(-0.0093)	0.2576	(+0.0115)	0.0128	(-0.0018)
		random keeping top 3	0.0415	(-0.0044)	0.2539	(+0.0078)	0.0133	(-0.0013)
	Proposed time overlap reranking method	without sliding window or keeping top	0.0392*	(-0.0067)	0.2691*	(+0.0230)	0.0143*	(-0.0003)
		with keeping top3	0.0425*$^{+}$	(-0.0034)	0.2771*$^{+}$	(+0.0310)	0.0153*$^{+}$	(+0.0007)
		with sliding window 3	0.0371*$^{+}$	(-0.0088)	**0.3000***$^{+}$	**(+0.0539)**	**0.0172***$^{+}$	**(+0.0026)**
		with keeping top 3 and sliding window 3	0.0417*$^{+}$	(-0.0042)	0.2901*$^{+}$	(+0.0440)	0.0162*$^{+}$	(+0.0016)

* The result of each user over (or loss under) the original algorithm is statistically significant at $p<0.01$(*)

+ The result of each user over (or loss under) the method of "without sliding window or keeping top" is statistically significant at $p<0.01$($^{+}$)

Two-tailed paired t test is used for the statistical significance test

acceptable precision loss (-7.9% using USG, -9.2% using LFBCA).

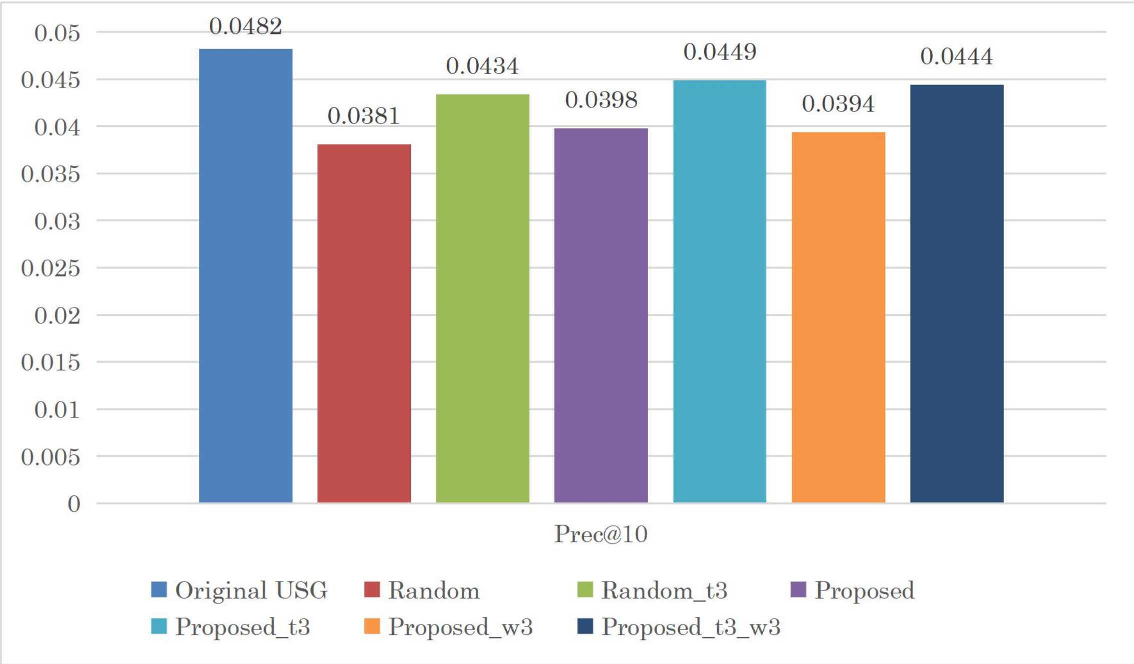


Figure 6: Precision@10 with **USG** Base Algorithm

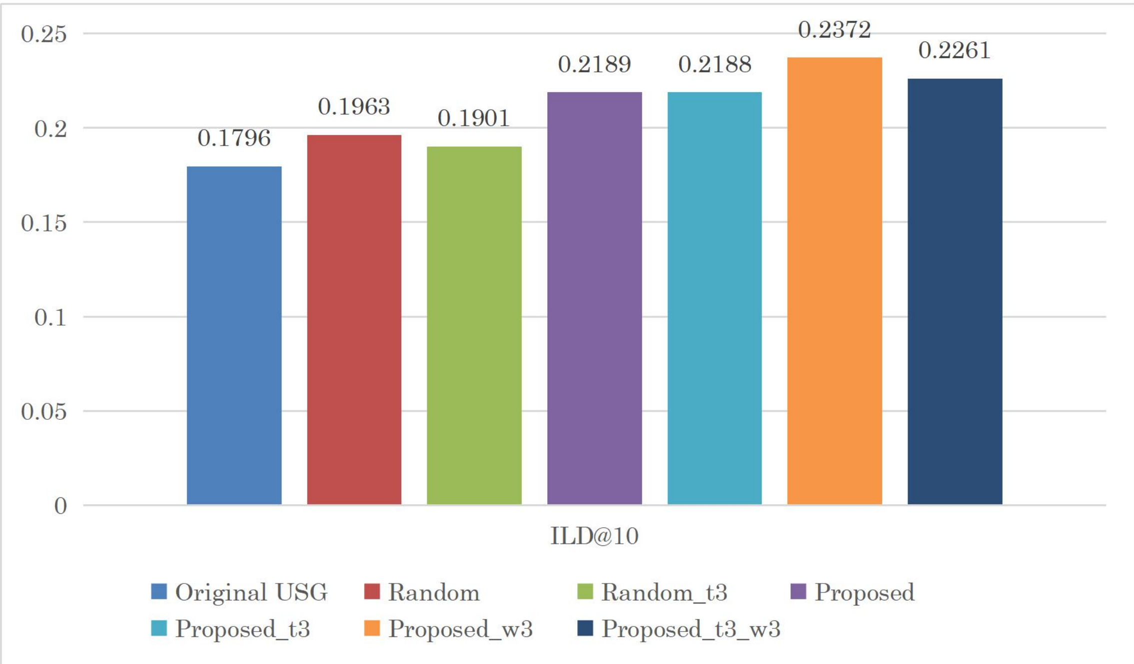


Figure 7: ILD@10 with **USG** Base Algorithm

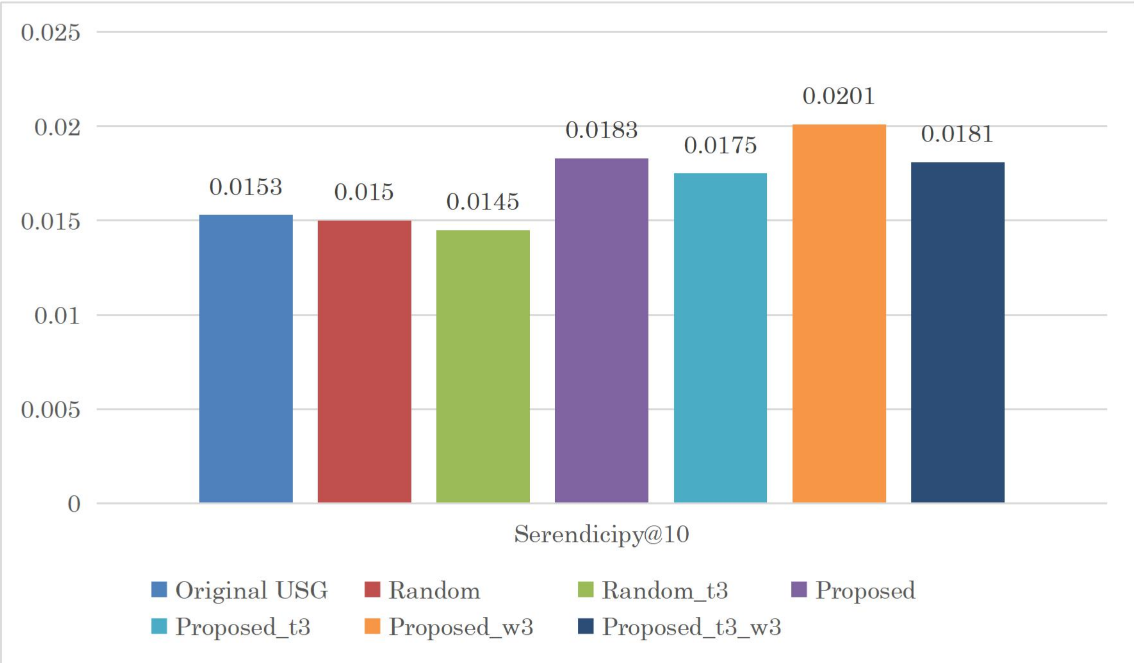


Figure 8: Serendicipy@10 with **USG** Base Algorithm

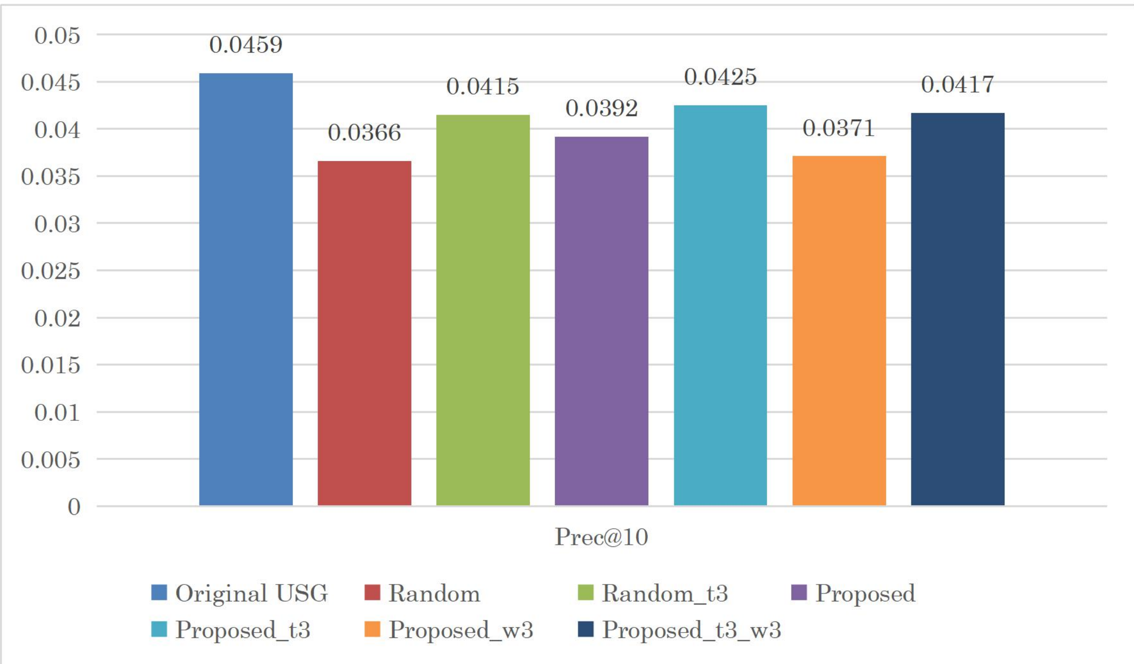


Figure 9: Precision@10 with **LFBCA** Base Algorithm

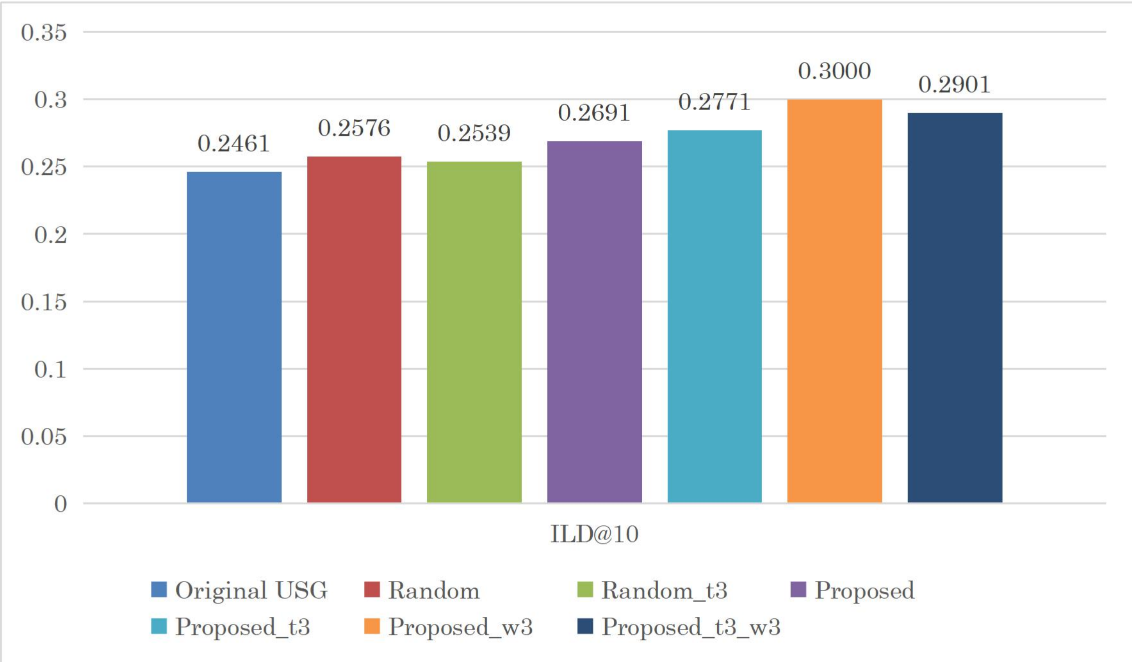


Figure 10: ILD@10 with **LFBCA** Base Algorithm

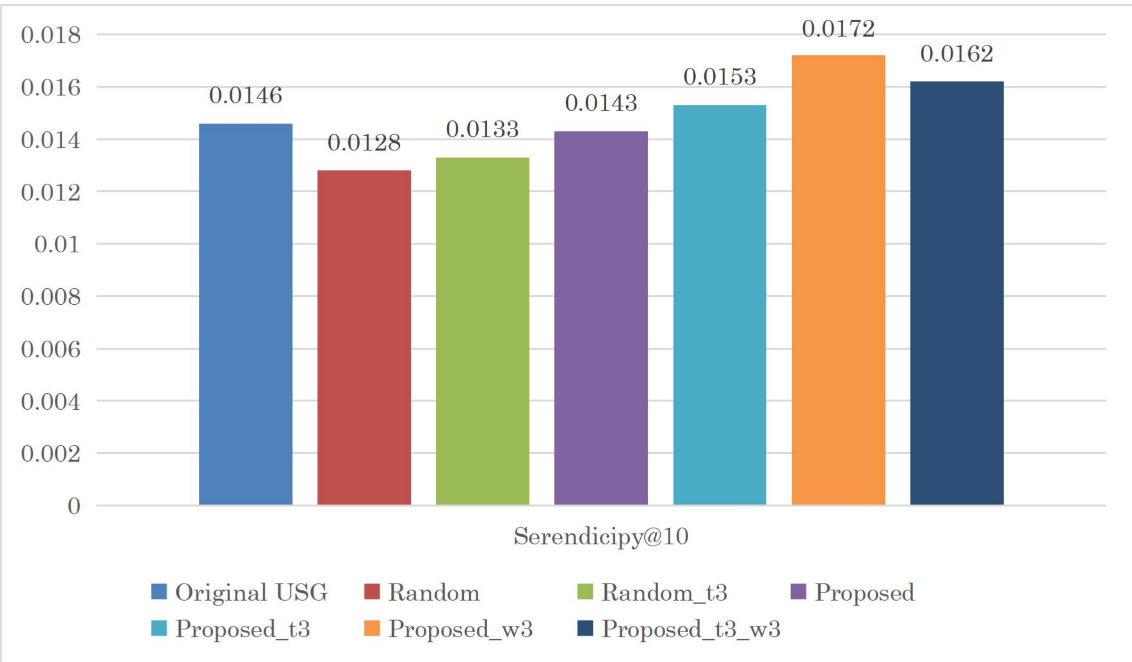


Figure 11: Serendicipy@10 with **LFBCA** Base Algorithm

5 Conclusion

In this paper, we introduce a new concept— *time diversity*; propose our "time-overlap reranking method" along with *sliding window* and *keeping top t* techniques to achieve time diversity to achieve user satisfaction. Experiment on real dataset, Gowalla, shows that our proposed method outperforms all other baselines in terms of *ILD@10* (25.9% increasement using USG [6] and 17.9% increasement using LFBCA). Meanwhile, the accuracy loss of the proposed method is tolerable (<10%). As future work, we shall increase the time slot from 24 to 48, separating work days and weekends, based on the assumption that users have different behavior patterns on work days and weekends, to further analyze users' time distribution. Besides, we shall consider improving our proposed method from the perspective of the reranking strategy. For instance, in place of the Keeping-top-t technique, we shall assign each POI a specific numerical probability of remaining. The probability is lower when it appears at the bottom of the candidate list, because the POIs at the top of the recommendation list have a higher accuracy, compared with those at the bottom.

References

[1] S. Harald. “Calibrated Recommendations.” in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 154–162.

[2] A. J. B. Chaney, B. M. Stewart, and . E. Engelhardt. “How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility”. in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp.224-232.

[3] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. “Improving Recommendation Lists through Topic Diversification.” in *Proceedings of the 14th International Conference on World Wide Web*, 2005, pp. 22–32.

[4] H. Jungkyu, and H. Yamana. “Geographical Diversification in POI Recommendation: toward Improved Coverage on Interested Areas.” in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 224–228.

[5] J. Zhang, and C. Chow. “Geosoca: Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations.” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 443–452.

[6] M. Ye, P. Yin, W. Lee, and D. Lee. “Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation.” in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information*, 2011, pp. 325–334.

[7] D. Zhang, M. Li, and C. Wang. “Point of Interest Recommendation with Social and Geographical Influence.” in *Proceedings of 2016 IEEE International Conference on Big Data*, 2016, pp. 1070–1075.

[8] Y. Shi, and W. Jiang. “Point-of-Interest Recommendations: Capturing the Geographical Influence from Local Trajectories”. in *Proceedings of 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications*, 2017, pp. 1122-1129.

[9] T. Shen, H. Chen, and W. Ku. “Time-aware location sequence recommendation for cold-start mobile users”. in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2018, pp. 484-487.

[10] X. Liu, Y. Liu, K. Aberer, and C. Miao. “Personalized point-of-interest recommendation by mining users' preference transition”. in *Proceedings of the*

22nd ACM international conference on Information & Knowledge Management, 2013, pp. 733-738.

[11] J. Zhang, C. Chow, and Y. Li. "Lore: Exploiting sequential influence for location recommendations". in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014, pp. 103-112.

[12] Z. Yao, Y. Fu, B. Liu, Y. Liu, and H. Xiong. "POI recommendation: A temporal matching between POI popularity and user regularity". in *Proceedings of 2016 IEEE 16th international conference on data mining*, 2016, pp. 549-558.

[13] R. Baral, and T. Li. "Maps: A multi aspect personalized poi recommender system". in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 281-284.

[14] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells. "Coverage, redundancy and size-awareness in genre diversity for recommender systems". in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 209-216.

[15] A. Barraza-Urbina, B. Heitmann, C. Hayes, and A. Carrillo-Ramos. "Xplodiv: An exploitation-exploration aware diversification approach for recommender systems". in *Proceedings of the Twenty-Eighth International Flairs Conference*, 2015.

[16] P. Zhao, and D. Lee. "How much novelty is relevant?: It depends on your curiosity". in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 315-324.

[17] F. Eskandanian, B. Mobasher, and R. Burke. "A clustering approach for personalizing diversity in collaborative recommender systems". in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 2017, pp. 280-284.

[18] M. Zanitti, S. Kosta, and J. Sørensen. "A user-centric diversity by design recommender system for the movie application domain". in *Companion Proceedings of the Web Conference 2018*, 2018, pp. 1381-1389.

[19] S. Lee, S. Kim, S. Park, and D. Chae. "A single-step approach to recommendation diversification". in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 809-810.

[20] P. Cheng, S. Wang, J. Ma, J. Sun, and H. Xiong. "Learning to recommend accurate and diverse items". in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 183-192.

[21] J. Wasilewski, and N. Hurley. "Intent-aware Item-based Collaborative Filtering for Personalised Diversification". in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, 2018, pp. 81-89.

[22] D.Valcarce, J. Parapar, and Á. Barreiro. “Combining top-n recommenders with metasearch algorithms”. in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 805-808.
[23] I. Benouaret, and D. Lenne. “A package recommendation framework for trip planning activities”. in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 203-206.
[24] H. Wang, M. Terrovitis, and N. Mamoulis. “Location recommendation in location-based social networks using user check-in data”. in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 374-383.
[25] Y. Liu, T. N. Pham, G. Cong, and Q. Yuan. “An experimental evaluation of point-of-interest recommendation in location-based social networks”. in *Proceedings of the VLDB Endowment*, 2017, pp. 1010-1021.
[26] H. Gao, J. Tang, X. Hu, and H. Liu. “Exploring temporal effects for location recommendation on location-based social networks”. in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 93-100.
[27] C. Cheng, H. Yang, I. King, and M. R. Lyu. “Fused matrix factorization with geographical and social influence in location-based social networks”. in *Proceedings of Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012 pp. 17-23.
[28] M. Ge, C. Delgado-Battenfeld, and D. Jannach. “Beyond accuracy: evaluating recommender systems by coverage and serendipity”. in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 257-260.
[29] J. Han, and H. Yamana. “Geographic Diversification of Recommended POIs in Frequently Visited Areas”. *ACM Transactions on Information Systems*, 2019, 38(1), pp. 1-39.
[30] Q. Yuan, G. Cong, Z. Ma, A. Sun, N. Magnenat- Thalmann. “Time-aware point-of-interest recommendation”. in *Proceedings of the 36th international ACM SIGIR Conference on Research and Development in Information Retrieval*, 2013, pp. 363-372.

Publication

Fan Mo, and Hayato Yamana. "Point of Interest Recommendation by Exploiting Geographical Weighted Center and Categorical Preference". in *Proceedings of IEEE International Conference on Data Mining (ICDM) Workshop on Big Social Media Data Management and Analysis (BSMDMA 2019)*, pp. 73-76. DOI: 10.1109/ICDMW.2019.00021.

Fan Mo, Huida Jiao, and Hayato Yamana. "Time Distribution based Diversified Point of Interest Recommendation". in *Proceedings of IEEE International Conference on Big Data Analytics (ICBDA 2020)*, 8 pages. (Submitted)