# Geographic Diversification of Recommended POIs in Frequently Visited Areas

JUNGKYU HAN, NAVER Corp., Korea
HAYATO YAMANA, Waseda University, Japan

In the personalized Point-Of-Interest (POI) (or venue) recommendation, the diversity of recommended POIs is an important aspect. Diversity is especially important when POIs are recommended in the target users' frequently visited areas, because users are likely to revisit such areas. In addition to the (POI) category diversity that is a popular diversification objective in recommendation domains, diversification of recommended POI locations is an interesting subject itself. Despite its importance, existing POI recommender studies generally focus on and evaluate prediction accuracy. In this article, geographical diversification (*geo-diversification*), a novel diversification concept that aims to increase recommendation coverage for a target users' geographic areas of interest, is introduced, from which a method that improves geo-diversity as an addition to existing state-of-the-art POI recommenders is proposed. In experiments with the datasets from two real Location Based Social Networks (LSBNs), we first analyze the performance of four state-of-the-art POI recommenders from various evaluation perspectives including category diversity and geo-diversity that have not been examined previously. The proposed method consistently improves geo-diversity (CPR(geo)@20) by 5 to 12% when combined with four state-of-the-art POI recommenders with negligible prediction accuracy (Recall@20) loss and provides 6 to 18% geo-diversity improvement with tolerable prediction accuracy loss (up to 2.4%).

CCS Concepts: • **Information systems** → Personalization; Information retrieval diversity; Recommender systems;

Additional Key Words and Phrases: POI recommendation, geographical diversity, LBSN, recommendation, POI, diversity

**ACM Reference format:**
Jungkyu Han and Hayato Yamana. 2019. Geographic Diversification of Recommended POIs in Frequently Visited Areas. *ACM Trans. Inf. Syst.* 38, 1, Article 1 (October 2019), 39 pages.
https://doi.org/10.1145/3362505

## 1 INTRODUCTION

As many Points-Of-Interest (POIs), places, or venues to visit can be located within a relatively small area in cities, in many cases it is difficult for users to know all the preferable POIs available. Personalized POI recommendation that recommends POIs preferred by a target user is one of the convenient tools in modern city life. The popularity of Location Based Social Networks (LBSNs) that allow users to share their visited POIs with their friends, and the pervasiveness of powerful

This work was carried out when the first author was a Ph.D student in Waseda University.

Authors' addresses: J. Han, NAVER Corp. Seongnam-si, Gyeonggi-do, Korea; email: jungkyu.han@navercorp.com; H. Yamana, Waseda University, Tokyo, Japan; email: yamana@waseda.jp.



1046-8188/2019/10-ART1 $15.00
https://doi.org/10.1145/3362505

mobile devices such as smart phones in the past decade, have made personalized POI recommendation feasible. Because POIs are located in a geographic space, users have to *move* at a proper *time point* to visit a POI. Extensive studies have shown that consideration of spatiotemporal characteristics such as locations and proper visiting times of POIs using conventional Collaborative Filtering (CF) further improves the performance of POI recommendation [22, 63].

Although performance is improved, those studies mainly focused on prediction accuracy, i.e., how many visited POIs in the test data are correctly recommended. Prediction accuracy is important but does not represent all aspects of recommendation quality. Other aspects (*beyond accuracy aspects* hereafter) such as diversity and novelty of recommended POIs are also important when considering recommendation quality. Diversification of recommended items is one of the popular beyond accuracy aspects in general recommendation models [16, 52, 78], and it is also important in POI recommendation. For instance, recommending restaurants from various preferable categories is better than recommending only French restaurants to the target user even if he or she likes French cuisine the most. Although recommending a list with many French restaurants will have increased prediction accuracy, because it has a higher chance of including POIs that the user may wish to visit, user preference for other preferred restaurant categories is sacrificed. To the best of our knowledge, there exist only a few POI recommender studies on the performance of beyond accuracy aspects, including diversity. Chen et al. [11] and Benouaret and Lenne [7] studied diversification of POI categories to improve information coverage of recommendation lists and diversity of trip packages, respectively.

Category diversification, described by the restaurant recommendation example, indicates a possible diversification of item (POI) categories appearing in recommended item (POI) lists. Although category diversification is the most popular approach in recommender system studies [11, 16, 78], geographic diversification (*geo-diversification* hereafter), diversification of POI locations that appear in a recommended POI list is also important. When our task is to recommend POIs in the areas frequently visited by the target user, it is preferable to recommend POIs in all sub-areas of the entire area than recommending POIs from only the most popular sub-area. Compared to a given user's infrequently visited areas, the user has many chances of revisiting their frequently visited areas. Thus, recommending preferable POIs from all sub-areas of interest to the target user is more appropriate than recommending POIs in a small and most-frequently visited sub-area. Geo-diversity is difficult to achieve by using existing exploitation methods of geographic features such as simply promoting POIs closely located to POIs visited by the target user. The method is likely to recommend POIs that are closely located to many of the user's previously visited POIs. As a consequence, the method is likely to recommend POIs located in the most visited sub-area of the user without providing diversity.

In this article, we introduce the concept of geo-diversification and then propose a novel geo-diversification method that improves the geo-diversity of existing POI recommenders. In the following experiments with two real LBSN datasets, we first show that it is difficult to improve geo-diversity with existing POI recommenders by evaluating the performance of basic CF methods and state-of-the-art POI recommenders. Thus, we evaluate the effectiveness of the proposed method by showing that geo-diversity is effectively improved in return of tolerable accuracy loss. Our contributions are threefold:

- Showing the performance of existing POI recommenders from a variety of evaluation perspectives.
- Proposing a novel geo-diversification concept and a method to instantiate the concept.
- Showing the effectiveness of the proposed method by extensive evaluation with real LBSN datasets.

The remainder of this article consists of five sections. In Section 2, related work is described. The POI recommendation problem we focus on and the concept of geo-diversity we pursue are explained in Section 3. Our proposed method is described in detail in Section 4 and is evaluated in Section 5. We conclude this article in Section 6.

## 2 RELATED WORK

We first describe studies on existing personalized POI recommenders and then explain methods for recommended item diversification.

### 2.1 Personalized POI Recommendation

**POI recommendation problems:** Generally, the POI recommendation problem is equivalent to a $k$-POI-selection problem. When user $u$, whose context set is $Ctx_u$, and the candidate POI set $C_{POI}$ which indicates the set of recommendable POIs to $u$ are given, a recommender selects the top-$k$ POIs that match $u$'s preferences from $C_{POI}$. By the geographic locations of the POIs in $C_{POI}$ and the contexts in $Ctx_u$, we can further classify the problem into several sub-problems.

Depending on the locations of the POIs included in $C_{POI}$, we can roughly define two sub-problems: POI recommendation for recall and POI recommendation for immediate visit. In the first sub-problem, the recommender does not care about the user's current location, but cares about the geographic distribution of the areas previously visited by the user [34, 63, 71, 72]. The recommender recommends $k$ POIs from the candidate POIs located in the areas previously visited by the user. This scenario represents a case where the user has less interest in an immediate visit to a POI. Instead the user wants to identify interesting POIs that can be visited later. For example, a user received a recommendation of POIs located nearby his/her workplace three days ago and then recalls the POIs when he/she is currently leaving the workplace.

In the second sub-problem, a target user's current location is important. The recommender assumes that a user can visit any area in a given region and will request recommendations within the visited area [34, 57]. The recommender selects $k$ POIs from the candidate POIs located within a close range from the user's current location such as within a $d$ km radius. This sub-problem focuses on the situation where the user is at an arbitrary location and requests a POI recommendation to identify nearby POIs. For instance, a user may want to know about interesting places located nearby when visiting Central Park in New York City.

User contexts are used for more accurate recommendation in either of the aforementioned sub-problems. Some algorithms [17, 48, 68, 69, 77] try to recommend more suitable POIs to visit at a requested time point or consider the users' visit time distribution over a 24-hour period to evaluate recommendation candidate areas. For example, pubs are likely not suitable answers to recommendation requests on a weekday morning. As another example, when we recommend restaurants to the user who frequently visits a given area in the morning, it is better to recommend POIs preferred by other users who also frequently visit the same area in the morning. Other algorithms [10, 20, 29, 75, 77] have studied the next POI recommendation problem to select the most feasible next $k$ POIs when the user's current visited POI is known. On the contrary, Feng et al. [19] studied an user prediction to predict users who will visit a given POI within some time period.

**Exploited features:** To improve the performance of personalized POI recommenders, additional features are incorporated in CF methods. Geographic proximity and visit time are POI-specific features while social relationships and contents of POIs, such as POI categories, user reviews, and sequential visit (interaction) patterns are conventional features that are also exploited for other recommendation tasks.

As POIs are located in geographical space and many are only available during certain time periods, spatiotemporal features such as geographic proximity and visiting times are the features that mostly differentiate POI recommendation from general item recommendation. Many studies promote POIs closely located to the POIs already visited by the target user (or that are located in the user's activity areas) [25, 35, 60, 63, 72] or promote POIs close to the user's current location [34]. Some algorithms exploit the user preference for POIs near a given POI when they calculate the user preference [42], based on the assumption that POIs closely located to each other must share common preferred features that are attractive to the user to some extent. Public visit-distribution-over-time is exploited to promote POIs that are suitable to visit at a given time point [17]. Individuals' temporal visit patterns to POIs or areas are also used to generate more personalized POI recommendations [22, 68].

Based on the observation that users visit the POIs visited by their friends to some extent [14], POIs visited by friends are promoted [36, 63, 72, 75]. POI contents such as POI categories and user reviews are also exploited. User preference to POI categories (e.g., French restaurants) can mitigate cold-start problems, especially when users visit infrequently visited areas where their visit logs are insufficient [3, 72]. Users' preferences for POIs can also be inferred from any reviews given by the users [5, 10, 26, 61, 73]. Observed sequential visit patterns of users are used mostly for the next POI recommendation [10, 20, 29, 75]. The aforementioned features are sometimes simultaneously considered with geographic context. For instance, public opinions about POIs near a given POI are considered to calculate the target user's preference to the given POI [73].

**Collaborative Filtering (CF):** Because the number of each user's observed visits to POIs is too low to correctly infer the user's preference to an arbitrary POI, CF methods are used to tackle the data sparsity problem. For memory-based CFs that load all user visit logs in the system memory, user-based CFs (MemUCFs) [63, 68, 71, 73] are popular methods. For model-based CFs that infer user preference models, matrix factorization–[12, 22, 23, 25, 35, 36, 39, 42, 55, 56, 61] and tensor factorization–[13, 29, 62, 77] based models are widely used. Topic model-based approaches [34, 54, 57, 64, 65, 72] are employed in studies that exploit POI contents such as POI categories or comments posted by users. Graph-based algorithms have also been studied [26, 28, 36, 40, 48, 59, 69] to exploit the influence propagation between two entities from arbitrary types such as users, POIs, visit times, and reviews. Recently, neural network-based models have been studied to incorporate complex features such as context affinity graphs [60], pictures [56], review texts [5], or sequential visit patterns of users [10, 76]. Since explicit user ratings of POIs are not available in many user visit logs, learning to rank methods [49] that use implicit visit data to learn user's pairwise preferences between any two given POIs have recently become popular methods [2, 13, 20, 29, 35, 38, 43, 66, 67, 75, 76] among model-based CF algorithms.

**Recommender evaluation:** In popular offline evaluations that examine user interaction logs instead of real user participation for evaluation, many recommender algorithm studies have evaluated their algorithms regarding prediction accuracy. Prediction accuracy indicates how many items in a given user's interaction (e.g., purchase) logs have actually been recommended to the user [30, 47]. *Prediction accuracy* indirectly indicates the user preference guessing performance of recommender algorithms. Although prediction accuracy is an important aspect for user satisfaction, studies repeatedly state that only focusing on accuracy cannot fully satisfy user and business demands [21, 44]. For instance, a set of popular preferable items includes the items purchased by the target user with high probability and therefore achieves increased prediction accuracy. Despite the high accuracy, the set also includes many already known items owing to their general popularity. In addition to prediction accuracy, *diversity* and *novelty* of recommended items are also popular aspects for user satisfaction [6, 51, 52, 78]. *Coverage* indicates the proportion of the number of items revealed to users to the number of all items in the system inventory [24, 30]. Coverage is an important aspect for business owners who register their items in a recommender system.

Although unified evaluation with various aspects can explain recommendation performance more comprehensively, almost the whole existing POI recommenders were evaluated on prediction accuracy. Even comprehensive studies, such as Liu et al. [41]'s evaluation of 12 state-of-the-art POI recommenders and Baral and Lee [4]'s role analysis of various POI recommendation features (e.g., social, temporal, spatial, and categorical aspect) focused on prediction accuracy. We do not exactly know the performance of existing POI recommenders regarding other evaluation aspects. In addition, despite the importance of POI location in POI recommendation, the effect of existing methods on the locations of recommended POIs has not been analyzed in detail. Geo-diversity as introduced in this article can be a metric used to evaluate a recommender's effect (how well the system considers location) on recommended POIs.

### 2.2 Recommendation Diversification

Diversification in recommender systems mainly indicates a heterogeneous item injection into a recommendation list. The injected items are selected based on adopted diversification objectives. The concept of result diversification has been introduced to recommendation systems from the field of information retrieval (IR) [52, 78]. Re-ordering methods [7, 8, 11, 16, 31, 51, 78] are popular diversification methods. An ordered list of $n$ items calculated by a base recommender that mostly focuses on prediction accuracy is re-ordered on the basis of a given diversification objective. Finally, the top-$k$ ($k < n$) items in the re-ordered list are presented to the user. Because popular diversification objectives of re-ordering methods are designed to measure item *category* similarity or difference, it is not clear how to apply them to *location* diversification. Feeling about geographically "close" or "far" varies over users or even cities. In some city, people think 1-km distance is not a far distance, but that may not true in other cities. Geographic distance is a more subjective perception than category difference that generally measured by a length of the shortest path between two categories on a taxonomy tree [78] or the-same-category-and-others classification [16]. As a result, simply applying a diversification objective designed for category diversification often does not get a satisfactory result. For instance, objective functions in a specific group of re-ordering methods [7, 8, 78] that try to maximize pairwise item distance can be applied to the location diversification objective by simply defining the "distance" as the geographic distance between two POIs. However, as a result, these methods simply recommend POIs that are located far from each other and do not adequately consider the target user's movement patterns.

Modification of the CF algorithms to handle diversity also has been proposed [1, 32, 45, 50, 53, 58]. For instance, Hurley et al. [32] proposed a personalized ranking method that incorporates pairwise dissimilarity between recommended items. Graph-based diversification methods [33] and clustering-based diversification methods [37, 74] that adopt graph and clustering algorithms, respectively, were also studied to recommend diversified items. However, these studies also did not consider the locations of items. In addition, because the "core" recommender algorithms must be modified, the cost incurred in applying these techniques to POI recommender systems that have already been deployed is high in comparison to simply applying re-ordering methods.

This article is an extended work of our previous study [27] about geo-diversification. Despite the concept of geographic diversification and a method applicable to add to existing POI recommender algorithms having been proposed in the previous study, the study included some generality limitations: (1) the proposed method was evaluated using only a single dataset and on a single existing POI recommender algorithm, and (2) the method ignored variation of user movement patterns, which are specific to each city where recommendations are calculated. More specifically, the method treats the user movement patterns as the same in all cities and assumes the patterns can be captured by a simple binary classification with globally static threshold values. This article improves the method in Referene [27] to be able to deal with different user movement patterns

Table 1. Notations

| Notation | Explanation | Notation | Explanation |
|---|---|---|---|
| $u, v, c, a$ | User $u \in U$, POI $v \in V$, check-in $c \in C$, v-area $a \in A$. | $U, V, C, A$ | The set of all users, POIs, check-ins, v-areas. |
| $cat$ | POI category $cat \in CAT$. | $CAT$ | The set of all POI categories. |
| $V_u^{Cand}$ | The set of recommendation candidate POIs for $u$. | $V_u^{Prof}$ | The set of visited POIs that appeared in $C_u^{Prof}$. |
| $CAT_v$ | The set of categories related to POI $v$. | $CAT_u^{Prof}$ | The set of categories for the POIs in $V_u^{Prof}$. |
| $C_u^{Prof}$ | The set (or list) of observed check-ins of user $u$. | $R_u^k$ | The set (or list) of $k$ recommended POIs for $u$. |
| $A_u^{Prof}$ | The set of v-areas included $u$'s active areas. | $V_u^{Test}$ | The set of POIs in $u$'s test set. |
| $x^{Loc}$ | The location of $x$. $x \in V$, or $x \in C$, or $x \in A$. | | |

for each city by adopting a visit probability distribution w.r.t. geographic distance. The proposed method in this article is evaluated with more datasets and existing POI recommender algorithms. In addition, the evaluation is carried out with additional evaluation metrics from various evaluation perspectives, such as novelty and coverage, to provide a clearer picture regarding the quality of POI recommendation.

## 3 PROBLEM DEFINITIONS

In Section 3.1, we first explain important terms such as POI, check-in, user profile, and v-area. Then the formal definition of the POI recommendation problem of interest is given. The concept of geo-diversity that is the goal of our proposed method is described in Section 3.2. Notations repeatedly used in the sections are summarized in Table 1.

### 3.1 POI Recommendation in Frequently Visited Areas

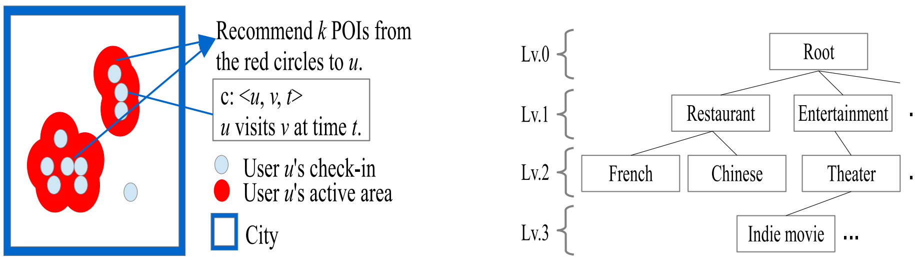
**POI:** A POI $v \in V$ is represented by a triplet of ID, POI categories, and location by ($v$, $CAT_v$, and $v^{Loc}$, respectively). $CAT_v$ indicates the set of POI categories related to $v$. For instance, if $v$ is "Happiness Café," then "Café" or "Restaurants" can be a member of $CAT_v$. $v^{Loc}$ indicates the location of $v$ on the geographic surface that is often represented by a latitude-longitude pair.

**Check-in and User profile:** A visit log or check-in $c$ consists of three attributes: a user, a POI, and the time of visit. User profile $C_u^{Prof}$ indicates the set of observed check-ins made by user $u$. Because $C_u^{Prof}$ is the set of visit logs, if $u$ visited POI $v$ twice at different time points, then $v$ appears twice in $C_u^{Prof}$. $V_u^{Prof}$ and $CAT_u^{Prof}$ indicate the set of visited POIs appearing in the check-ins in $C_u^{Prof}$ and the set of categories for the POIs appearing in $V_u^{Prof}$, respectively.

**POI recommendation in frequently visited areas:** For each user $u \in U$, our POI recommendation problem is defined as recommending $k$ unvisited POIs from the set of recommendation candidate POIs. Equation (1) provides a formal expression of the recommended POI set $R_u^k$. Although notation $R_u^k$ is defined to indicate a *set*, for easy explication, we slightly abuse $R_u^k$ to indicate a recommended POI *list*, too, because recommended POIs are generally presented to users in the form of ordered lists,

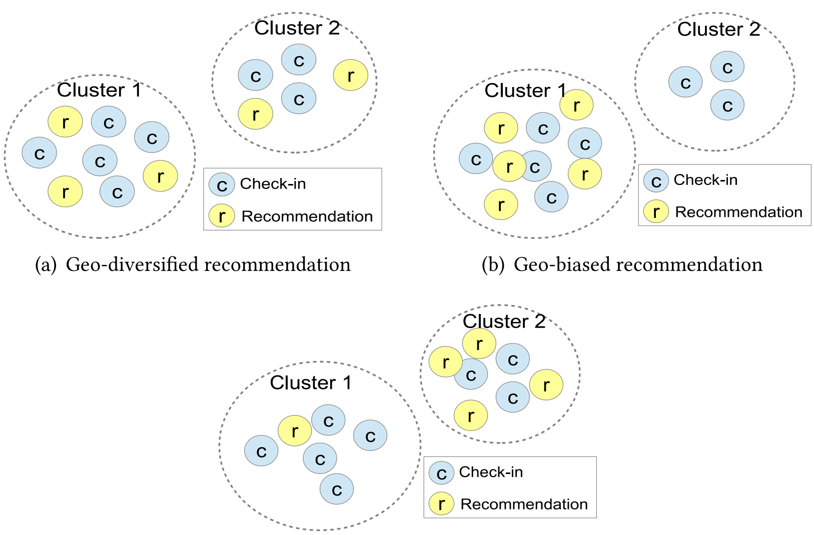
$$R_u^k = \left\{ v \in V \;\middle|\; v \in \left(V_u^{Cand} \setminus V_u^{Prof}\right) \right\} \quad \text{and} \quad \left|R_u^k\right| = k. \tag{1}$$

As we are interested in recommendation of POIs located in each user's frequently visited areas, a special case of the first sub-problem described in Section 2.1 arises, and it is desirable that only the POIs located in $u$'s frequently visited areas are included in $V_u^{Cand}$. To formally define frequently visited areas, we first define *visitable areas (v-areas)* of the user and then represent the frequently visited areas of the user by a subset of the v-areas. v-Areas of a given user is defined as $th_{far}$-km-radius-circles whose center is equivalent to each check-in of the user. v-Area $a_{u,c}$ indicates a viable reach of user $u$ from an observed visit location $c^{Loc}$. Because an arbitrary check-in pair of the




(a) POI recommendation (b) Category tree

Fig. 1. Target problem and Category tree.




(a) Geo-diversified recommendation (b) Geo-biased recommendation

(c) Less Geo-diversified recommendation

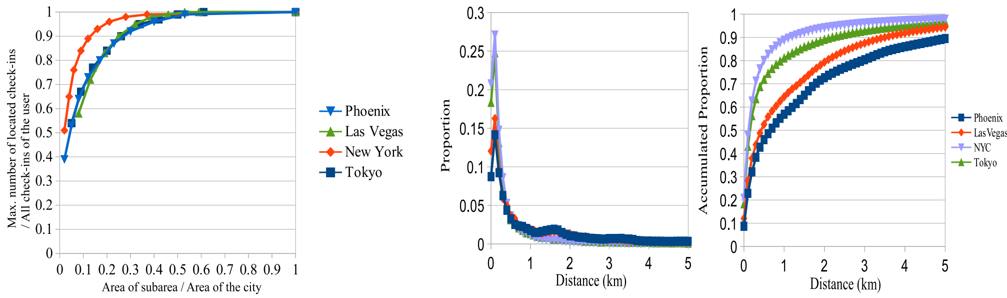
Fig. 2. Geographical diversity examples.

same user can be located closer than $2 \cdot th_{far}$ km, two v-areas of user $u$ can be overlapped. Active areas of user $u$, $A_u^{Prof}$, indicates frequently visited areas of $u$. $A_u^{Prof}$ consists of v-areas that occupy the minimum sized area in total while contain most check-ins of $u$. We describe the calculation of $th_{far}$ and $A_u^{Prof}$ in detail in Section 4.1. We define $V_u^{Cand}$ as the set of all POIs located in $A_u^{Prof}$ (Equation (2)). In summary, our problem is equivalent to generating recommendation list $R_u^k$ by extracting appropriate POIs from the POIs located in $u$'s active areas. Figure 1(a) shows an example of our POI recommendation problem,

$$V_u^{Cand} = \left\{ v \in V \;\middle|\; (\exists a_{u,c}) \left[ v^{Loc} \in a_{u,c} \wedge a_{u,c} \in A_u^{Prof} \right] \right\}. \tag{2}$$

### 3.2 Geographical Diversity

We describe the goal of geographical diversification (geo-diversification) in detail. In this article, geo-diversity indicates a type of similarity for the location distribution over a geographic surface between recommended POIs and the observed check-ins of the target user. In other words, geo-diversity in this article does not mean just recommending POIs located in different locations. It instead indicates recommended POIs based on locations visited by the target user. Figure 2 shows




(a) Maximum number of user check-in w.r.t. sub-area size

(b) Check-in frequency ratio

Fig. 3. Check-in concentration in small area of the city and check-in frequency w.r.t. distance.

a good example and two poor examples of geo-diversity. In each example, the user's observed check-ins are roughly divided into two different-size clusters on a geographic surface, and we recommend five POIs to the user. Figure 2(a) achieves better geo-diversity than 2(b) in that the recommended POIs are concentrated in the big cluster and the example in 2(c) in that more of the recommended POIs are located at nearby locations in the smaller cluster.

Geo-diversity is more effective when we recommend POIs from users' frequently visited areas. Compared to the case of requesting nearby POIs for immediate visit, the movement distance is a less severe barrier in users' frequently visited areas, because they know the areas well and are likely to revisit them. In such a case, it is more desirable to recommend POIs that geographically cover the frequently visited area as much as possible to increase the possibility of satisfying user requirements. The proportionality of the recommended POIs to the nearby user check-ins in geo-diversity allows the POI recommender to provide the user with more information about the more frequently visited areas.

## 4 GEOGRAPHICAL DIVERSIFICATION ALGORITHM

In this section, we propose a geo-diversification method to balance accuracy and geo-diversity. First, a user-active-area-selection algorithm to calculate the candidate POIs is proposed. After that an overview of the proposed diversification algorithm is described. Finally, the geo-diversity measure adopted in the objective function of the proposed diversification algorithm is described in detail.

### 4.1 Active Area Selection

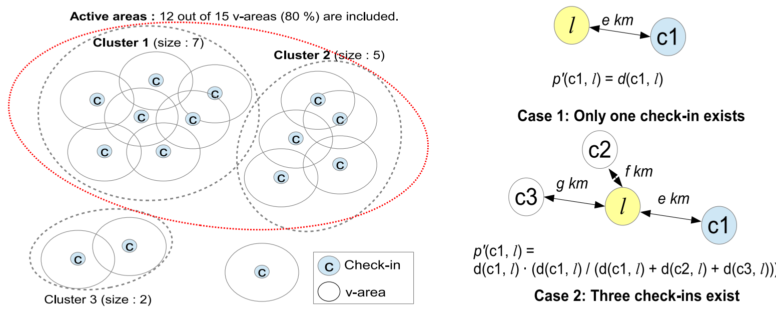
We first identify the target user's frequently visited areas $A_u^{Prof}$ to calculate candidate POI set $V_u^{Cand}$. To carry out the task, we exploit a Pareto principle [18] observed from user check-ins. In other words, user check-ins are concentrated on relatively small frequently visited areas compared to the area of entire city. For example, Figure 3(a) represents the check-in ratio of the user that located in top $x\%$ of $500 \times 500$ m sub-areas [27] from the sub-area list ordered by the user's visit density in descending order. The visit density is calculated for each user by Adaptive Density Kernal [72] with check-in data described in Section 5.1. Over 80% of visits of each user are concentrated on less than 20% of the entire area of each city. Similar observations are found in other studies on user movement pattern [12, 14], and in part, that can be explained by visit concentration on frequently visited areas [14]. Based on the observation, we consider the minimal sized areas containing about 80% of check-ins of a given user as the frequently visited areas of the user.

Our proposed active area selection algorithm agglomerates overlapped *visitable areas (v-areas)* of a given user to formulate clusters of the v-areas and then extracts minimal number of top big clusters containing about 80% of the user's check-ins in total. v-Areas not overlapped to any of other v-areas are ignored, because the check-ins in the center of the areas are not likely to be generated by frequent visit of the user. We regard the areas defined by the union of the v-areas in the extracted clusters as the active areas of the user. Algorithm 1 represents the details of the selection algorithm and Figure 4(a) shows an example of selected v-area clusters for user $u$ (active areas $A_u^{Prof}$).

As described in Section 3.1, v-area $a_{u,c}$ of user $u$ is defined as a $th_{far}$-km-radius-circle whose center is equivalent to the location of $u$'s check-in $c$. The value of $th_{far}$ may be different based on individuals or cities. For example, some users prefer longer distance visits than others or longer distance visits are more frequent in some cities than other cities owing to different cultural and traffic environments. We calculate $th_{far}$ based on user movement patterns observed from the evaluation datasets described in Section 5.1 to consider the differences among various cities: Phoenix, Las Vegas, New York, and Tokyo. Figure 3(b) shows the proportion of check-ins ($y$-axis) whose distance from their geographically closest check-ins made by the same user fall within the $[x, x + 0.1)$ km range. The graph can be considered as a discrete probability distribution of an $x$ distance movement in each city. $th_{far}$ is taken the minimum distance ($x$ km) that the aggregate ratio of the nearest check-in existence becomes greater than 0.8 to exploit the observed Pareto principle shown in Figure 3(a). Since the probability distribution follows power-law distribution from 0.1 km

**ALGORITHM 1:** Active area selection algorithm (aaSelect)

**Input:** $u, C_u^{Prof}, th_{far}$
**Output:** $A_u^{Prof}$
$G(V = C_u^{Prof}, E = \phi)$; $A_u^{Prof} \leftarrow \phi$; List $L \leftarrow \phi$; $a \leftarrow 0$;
**for** $c_i, c_j \in G.V$ **do**
  **if** $c_i \neq c_j \wedge distance(c_i^{Loc}, c_j^{Loc}) \leq 2 \cdot th_{far}$ **then**
    $G.E \leftarrow G.E \cup \{e_{ij}, e_{ji}\}$;
  **end if**
**end for**
**for** $c_i \in G.V$ **do**
  $G_i(V, E) \leftarrow BFS(c_i)$; (Breadth First Search from $c_i$)
  $G.V \leftarrow G.V \setminus G_i.V$;
  $G.E \leftarrow G.E \setminus G_i.E$;
  **if** $|G_i.V| > 1$ **then**
    $L.add(G_i)$;
  **end if**
**end for**
$SortByTheGraphOrderInDescent(L)$;
**for** $i = 1$ to $L.size$ **do**
  $G_i \leftarrow L[i]$;
  $A_u^{Prof} \leftarrow A_u^{Prof} \cup \{a_{u,c} | (\exists c)[c \in G_i.V]\}$;
  $a \leftarrow a + |G_i.V|$;
  **if** $\frac{a}{|C_u^{Prof}|} \geq 0.8$ **then**
    break;
  **end if**
**end for**




(a) Active area selection for user $u$. (b) Weighted count

Fig. 4. Active area selection and weighted counts of a closely located POI.

distance, we can regress them in the form of Equation (3) by using a regression technique [63] and calculate $th_{far}$ by Equation (4). $\gamma$ in Equation (4) is the accumulated proportion of check-ins whose distance from their closest check-ins is less than 0.1 km. The calculated $\alpha - \beta - \gamma$ triplets are (0.108, −0.782, 0.072), (0.126,−0.816, 0.102), (0.222, −1.122, 0.206), and (0.205, −1.102, 0.181), and $th_{far}$s are 4.3, 2.9, 0.8, and 1.1 for Phoenix, Las Vegas, New York, and Tokyo, respectively (listed values are the average values calculated for each validation of 5-fold cross validation, as described in Section 5.1),

$$p_{check-in}(x) = \begin{cases} \alpha \cdot (10x - 1)^{\beta} & x \geq 0.1 \\ 0 & x < 0.1 \end{cases}, \tag{3}$$

$$th_{far} = \underset{\theta}{\operatorname{argmin}} \left( \int_{0.1}^{\theta} p_{check-in}(x)dx + \gamma \geq 0.8 \right). \tag{4}$$

## 4.2 Diversification Algorithm Overview

We propose a reordering-based recommender to instantiate geo-diversification. Reordering-based recommenders reorder a given base recommendation list of size $n$ calculated by a base recommender according to reordering criteria and then recommend the top-$k$ ($k < n$) items from the re-ordered list as the final recommendation. We adopt reordering, as it decouples diversification from the base recommender, and therefore the algorithm is flexible enough to apply to already deployed POI recommender systems.

Algorithm 2 shows the recommendation procedure of our proposed method. *BaseRec* and $n$ indicate the base recommender and the number of POIs recommended by the base recommender, respectively. Like most of the reordering algorithms described in Section 2.2, reordering is carried out by greedy selection; the item (POI) that achieves the maximum increment of the diversification objective (reordering criterion) at a given selection trial is selected from the base item list and is then placed at the tail of the final recommendation list [16, 51, 78].

Equation (5) represents the diversification objective of our proposed method. $rel(u, i)$ indicates the relevance score of POI $i$ to user $u$, and $div_{geo}(u, i, R_u^k)$ (Equation (6)) indicates the $PR_u^{geo}(R_u^{k+1})$ (details are described in Section 4.3.2) increment by adding $i$ to $R_u^k$ at a given time point. $PR_u^{geo}$ metric measures a kind of geographic location distribution similarity between POIs in the recommendation list $R_u^k$ and the observed check-ins of the target user $C_u^{prof}$. For $rel(u, i)$, we use the score calculated for the user $u$ and POI $i$ pair by the base recommender. Since the proposed method pursues the aforementioned two objectives simultaneously, it tries to recommend the target user's preferable POIs whose location distribution is similar to his/her visit pattern over

**ALGORITHM 2:** Geo-diversification algorithm (geoDiv)

**Input:** $u, C_u^{Prof}, V_u^{Cand}, k, BaseRec, n$
**Output:** $R_u^k$
  $poiList \leftarrow BaseRec.getPOIRec(u, n, V_u^{Cand})$;
  $R_u^k \leftarrow \phi$;
  **for** 1 to $k$ **do**
    $max \leftarrow 0$;
    $l \leftarrow \phi$;
    **for** each POI $i \in poiList$ **do**
      $obj \leftarrow obj_{GeoD}(u, i, R_u^k)$; (Equation 5)
      **if** $obj > max$ **then**
        $max \leftarrow obj$;
        $l \leftarrow i$;
      **end if**
    **end for**
    $poiList \leftarrow poiList \setminus \{l\}$;
    $R_u^k \leftarrow R_u^k \cup \{l\}$;
  **end for**

geographic areas. $\lambda \in [0.0, 1.0]$ denotes the diversification weight to control a balance between the two objectives,

$$obj_{GeoD}\left(u, i, R_u^k\right) = rel(u,i)^{1-\lambda} \cdot div_{geo}\left(u, i, R_u^k\right)^{\lambda}, \tag{5}$$

$$div_{geo}\left(u, i, R_u^k\right) = max\left(0, PR_u^{geo}\left(\{i\} \cup R_u^k\right) - PR_u^{geo}\left(R_u^k\right)\right). \tag{6}$$

### 4.3 Geo-diversity Measurement

Geo-diversity as described in Section 3.2 indicates a location distribution similarity between recommended POIs and the target user's check-ins over a geographic surface, so to achieve high geo-diversity, our proposed method tries to meet the following two conditions: **(R1)** the recommended POIs are located closer to the POIs appearing in user $u$'s check-ins. In addition, **(R2)** the number of recommended POIs in an arbitrary location is proportional to the number of $u$'s check-ins that are located nearby. To judge whether the recommended POIs satisfy (**R1**) and (**R2**), we adopt $PR^{geo}$, a modified $PR^{cat}$ (Equation (7)) measure to calculate the location distribution similarity of two location sets. We first describe $PR^{cat}$ and then explain how $PR^{cat}$ is modified to $PR^{geo}$.

*4.3.1 Category Distribution Similarity Measurement.* $PR^{cat}$ was introduced by Dang et al. [16]. $PR^{cat}$ is used to measure a type of distribution similarity between the distribution of an aspect of interest in user profile $C_u^{Prof}$ and the distribution of the aspect in recommended list $R_u^k$. When the proportion of the aspect appearance in $R_u^k$ is more similar to that of the aspect appearance in $C_u^{Prof}$, a higher value is assigned to $PR^{cat}@k$. In the recommendation context, a category is usually used as a popular aspect of interest. For instance, if 70% of check-ins in $C_u^{Prof}$ are visits to POIs in category $cat_1$ and the remaining 30% are visits to POIs in category $cat_2$, then a recommendation list that contains 65% POIs from $cat_1$ and 35% from $cat_2$ has a higher $PR^{cat}@k$ value than a recommendation list that contains 30% POIs from $cat_1$ and 70% from $cat_2$. $p(i, C_u^{Prof})$ and $p(i, R_u^k)$ in Equation (8) and Equation (9) indicate the frequency proportion of category $i$ in the datasets $C_u^{Prof}$ and $R_u^k$, respectively. For example, if 30% of check-ins in $C_u^{Prof}$ are visits to POIs in $cat_1$, then $p(cat_1, C_u^{Prof}) = 0.3$. $n_{NR}$ in Equation (8) indicates the number of non-relevant POIs that are not included in any of the user visited categories,

$$PR_u^{cat}@k = 1 - \frac{DP_u^{cat}@k}{Ideal\_DP_u^{cat}@k}, \tag{7}$$

$$DP_u^{cat}@k = \sum_{i \in CAT_u^{Prof}} f_i \cdot \left(p\left(i, C_u^{Prof}\right) - p\left(i, R_u^k\right)\right)^2 + \frac{1}{2}\left(\frac{n_{NR}}{k}\right)^2$$

$$\text{where} \quad f_i = \begin{cases} 1 & p\left(i, C_u^{prof}\right) \geq p(i, R_u^k) \\ 0 & otherwise \end{cases}, \tag{8}$$

$$Ideal\_DP_u^{cat}@k = \sum_{i \in CAT_u^{Prof}} \left(p\left(i, C_u^{Prof}\right)\right)^2 + \frac{1}{2}. \tag{9}$$

*4.3.2 Location Distribution Similarity Measurement.* In this subsection, we describe how $PR^{cat}@k$ is modified to $PR^{geo}@k$ to calculate the location distribution similarity of two location sets. Instead of using the user visit frequency of each POI category as the reference distribution for recommended POI diversification, we use the check-in locations of the user as the reference distribution. To achieve high $PR_u^{geo}$, a "sufficient amount" of recommended POIs should be geographically "closely located" to each check-in location. $p(i, C_u^{Prof})$ and $p(i, R_u^k)$ in Equation (8) and Equation (9) are modified to measure the two aforementioned properties. Because user $u$ can make multiple check-ins to the same POI, closely placing the same number of "distinct" recommended POIs to each check-in location makes the number of recommended POIs in an area become approximately proportional to the number of $u$'s visits made to that area. When user $u$ has $|C_u^{Prof}|$ check-ins and $k$ POIs are recommended to $u$, in the ideal case, $k/|C_u^{Prof}|$ distinct recommended POIs are closely positioned to each check-in location. To achieve the aforementioned idea, $p(i, C_u^{Prof})$ is replaced by $Req_u$ (Equation (13)) so that $Req_u$ distinct recommended POIs are closely located to each check-in location in user profile $C_u^{Prof}$. As a result, Equation (8) and Equation (9) are replaced by Equation (11) and Equation (12), respectively.

The remaining questions that we have to answer are "how to measure closeness" and "how to count the distinct close POIs" that are related to the $p(i, R_u^k)$ modification to $p(c, R_u^k)$ in Equation (14).

**Closeness measurement:** The degree of "closeness" about a given geographic distance is different based on individuals or cities. Adopting a similar calculation (Equation (4)) for $th_{far}$, we measure the geographic closeness for each city based on user movement patterns observed in Figure 3(b). We define the closeness at given distance $x$ as "1 minus the accumulated check-in probability from distance 0 to $x$," because we think that the closeness is proportional to the probability of "not moving a shorter distance than $x$." As a result, the degree of closeness is calculated by Equation (16).

We do not consider distances of more than $th_{far}$ km. In addition, the distributions are normalized to make the accumulated probability equal to 1 when $x$ equals $th_{far}$ km. This policy is taken to reduce noise, because negligible probabilities exist at more than $th_{far}$ km distances and most check-ins have at least negligible closeness values to most of the recommended POIs if we consider POIs located further away than $th_{far}$ km.

**Quantity measurement:** $p(c, R_u^k)$ calculates the fraction of distinct POIs in $R_u^k$ that are closely located to a given check-in $c$. Because the number of recommended POIs is usually smaller than the number of user $u$'s check-ins, the weight (closeness degree) of a single recommended POI closely located to multiple check-ins is divided based on the closeness to each check-in and is shared by each check-in. As shown in Case 1 of Figure 4(b), when recommended POI $l$ is closely located to check-in $c_1$ only, the weight of $l$ for $c_1$ is equal to the closeness of $l$ to $c_1$ ($d(c_1, l)$ in Equation (16)). When $l$ is closely located to three check-ins $c_1$, $c_2$, and $c_3$ simultaneously, as shown in Case 2 of Figure 4(b), the value of $d(c_1, l)$ is divided to calculate $p(c_1, R_u^k)$, because $l$ is shared by the three check-ins. The amount of the divided value for each check-in is proportional to the closeness of each check-in to $l$. For example, the value of $d(c_1, l)$ divided by the sum of $d(c_1, l)$, $d(c_2, l)$, and $d(c_3, l)$ contributes to $p(c, R_u^k)$ (Equation (15)). $n_{NR}$ in Equation (11) indicates the number of recommended POIs that are not closely located to any check-in $c \in C_u^{Prof}$. In other words, $n_{NR}$ indicates the number of recommended POIs $l$ that satisfy $d(c, l) = 0$ for all $c \in C_u^{Prof}$,

$$PR_u^{geo}@k = PR_u^{geo}\left(R_u^k\right) \quad \text{and} \quad PR_u^{geo}@k = 1 - \frac{DP_u^{geo}@k}{Ideal\_DP_u^{geo}@k}, \tag{10}$$

$$DP_u^{geo}@k = \sum_{c \in C_u^{Prof}} f_c \cdot \left(Req_u - p\left(c, R_u^k\right)\right)^2 + \frac{1}{2}\left(\frac{n_{NR}}{k}\right)^2, \quad \text{where} \quad f_c = \begin{cases} 1 & Req_u \geq p\left(c, R_u^k\right) \\ 0 & otherwise \end{cases}, \tag{11}$$

$$Ideal\_DP_u^{geo}@k = \sum_{c \in C_u^{Prof}} \left(Req_u^2\right) + \frac{1}{2} = \left(\left|C_u^{prof}\right| \cdot Req_u^2 + \frac{1}{2}\right), \tag{12}$$

$$Req_u = \frac{k}{|C_u^{prof}|}, \tag{13}$$

$$p\left(c, R_u^k\right) = \sum_{l \in R_u^k} p'(c, l), \tag{14}$$

$$p'(c, l) = d(c, l) \cdot \frac{d(c, l)}{\sum_{c' \in C_u^{prof}} d(c', l)}, \tag{15}$$

$$d(c, l) = \begin{cases} 1 - \frac{\int_{0.1}^{dist} p_{check-in}(x)dx + \gamma}{\int_{0.1}^{th_{far}} p_{check-in}(x)dx + \gamma} & th_{far} \geq dist \\ 0 & dist > th_{far} \end{cases}, \quad \text{where} \quad dist = \text{km-distance}(c^{Loc}, l^{Loc}). \tag{16}$$

**Elimination of some check-ins from $C_u^{Prof}$:** Although $C_u^{Prof}$ in Table 1 indicates the set of all observed check-ins of user $u$, in this article, only the check-ins in $C_u^{Prof}$ that are located less than $th_{far}$ km distance to at least one POI in the candidate recommendation POI set $V_u^{Cand}$ are used to calculate $PR_u^{geo}$ and $CPR_u^{geo}$ (Section 5.1.2). The reason for the elimination of some check-ins is that we are recommending POIs located in frequently visited areas that usually do not cover the entire area of the city.

## 5 EVALUATION

In this section, we evaluate our proposed method from various evaluation perspectives. The datasets used, the evaluation procedure, and the adopted evaluation metrics are described as parts of the experimental setup in Section 5.1. Then the performances of existing POI recommender algorithms are evaluated in Section 5.2. As the proposed method re-orders the POIs in the recommended list calculated by a base recommender, a preliminary performance analysis on possible base POI recommenders is carried out to analyze the performance of the proposed method more clearly. Finally, in Section 5.3, the proposed method is evaluated and discussed in detail.

Table 2. Data Statistics

| City | Dataset | Users | POIs | Check-ins | Sparsity | $th_f$ | Friend link |
|---|---|---|---|---|---|---|---|
| Phoenix | Yelp | 20,713 | 24,386 | 636,228 | 0.0013 | 4.3 | Avail. |
| Las Vegas | Yelp | 23,145 | 7,720 | 268,289 | 0.0015 | 2.9 | Avail. |
| New York | Foursquare | 7,041 | 17,417 | 759,862 | 0.0062 | 0.8 | N/A |
| Tokyo | Foursquare | 4,350 | 9,411 | 202,894 | 0.0050 | 1.1 | N/A |

### 5.1 Experimental Setup

*5.1.1 Dataset and Evaluation Procedure.* **Dataset:** The data of the four cities from two real LBSNs are used in our experiment: Phoenix and Las Vegas in the U.S. from the Yelp Dataset Challenge[1] and New York in the U.S. and Tokyo in Japan from the Foursquare[2] dataset.[3] We select Phoenix and Las Vegas, because the two cities have the most user visit logs than other cities in the Yelp dataset. We collected publicly available Foursquare data by using Twitter API[4] and Foursquare API.[5] Because we could not collect Foursquare friend information, friend links are not available for consideration in the Foursquare dataset. Data are usually preprocessed to reduce noise in evaluation. Many POI recommendation studies filter out users and POIs related to small number of visit logs (usually from 5 to 20 logs) [43, 67, 68, 75]. For instance, GeoBPR [67], one of the base recommenders adopted by our proposed method, evaluated their algorithm with the data of the users who have at least 20 visit logs and the POIs visited by at least five users. We preprocessed the data to only evaluate the users who have at least 10 visit logs and to POIs visited by at least five users to evaluate more users who have small sized histories. Basic statistics of the data after preprocessing are shown in Table 2.

The dataset provides a hierarchical category of POIs similar to Figure 1(b). The categories at the Lv.2 height in the category tree are used to indicate a given POI's categories. We also assume an Lv.2 height category tree when a treestructure-based operation such as a category similarity calculation is needed.

**Evaluation procedure:**

- We carried out fivefold cross validation to evaluate a given algorithm. For each validation,
  - 80% of the check-ins of each user are used for training (Model learning: 70%, Tuning: 10%).
  - The remaining 20% are used for testing.
  - In the test, the algorithm is evaluated with the metrics from various perspectives:
    - Prediction accuracy, diversity, novelty, coverage, and joint metric. (Table 3)
- The results are reported as the averages of all validations.

In the test, POIs are recommended from candidate POI set $V_u^{Cand}$ (Equation (2)) for each user $u$. In addition, some evaluation metrics require ground-truth POIs for each $u$. The ground-truth POIs are identical to the POIs in $V_u^{Cand}$ that satisfy two conditions simultaneously: (1) the POIs appear in $u$'s test set and (2) do not appear in $u$'s training set.

[1] https://www.yelp.com/dataset/challenge.
[2] https://foursquare.com/.
[3] Available at https://drive.google.com/drive/folders/1Gpjo8TPIzisR23CSx9TEhbs2xiUpuARf?usp=sharing.
[4] https://developer.twitter.com/en/docs.
[5] https://developer.foursquare.com/.

Table 3. Evaluation Metrics

| Evaluation Perspective | Metric (Abbreviation) | Range: [worst, best] | Explanation | Target POIs* |
|---|---|---|---|---|
| Prediction accuracy (Accuracy) | *Precision@k* [30] (*P*) | [0.0, 1.0] | The fraction of recommended ground-truth (GT) POIs among the $k$ recommended POIs. | Hit only |
| | *Recall@k* [30] (*R*) | [0.0, 1.0] | The fraction of recommended GT POIs by $k$ POI recommendation list over the total amount of GT POIs. | Hit only |
| Diversity | $ILD^{cat}@k$ [78] ($IL^c$) | [0.0, 1.0] | The average category distance between two POIs in the $k$ POI recommendation list. | All |
| | $ILD^{geo}@k$ [78] ($IL^g$) | [0.0, Max. possible km length of each city] | The average geographic distance (km) between two POIs in the $k$ POI recommendation list. | All |
| | $SRecall^{cat}@k$ [70] ($SR^c$) | [0.0, 1.0] | The fraction of the GT POI categories in $k$ POI recommendation list over the total amount of GT categories. | All |
| | $SRecall^{geo}@k$ [70] ($SR^g$) | [0.0, 1.0] | The fraction of the GT active areas in $k$ POI recommendation list over the total amount of GT active areas. | All |
| | $CPR^{cat}@k$ [16] ($CP^c$) | [0.0, 1.0] | The degree of similarity between the category distribution in $k$ POI recommendation list and the category distribution in $C_u^{Prof}$. | All |
| | $CPR^{geo}@k$ ($CP^g$) | [0.0, 1.0] | The degree of similarity between the location distribution in $k$ POI recommendation list and the location distribution in $C_u^{Prof}$. | All |
| Novelty | *EPC@k* [45] (*EP*) | [0.0, 1.0] | The average degree of non-popularity of recommended GT POIs by $k$ POI recommendation. Higher value indicates that the recommended GT POIs are less popular. | Hit only |
| Catalog coverage (Coverage) | *AggDiv@k* [21] (*AD*) | [0, $\|V\|$] | The number of distinct retrieved GT POIs by $k$ POI recommendations for all users. | Hit only |
| Joint Metric (Accuracy & Diversity) | $\alpha$ - $NDCG^{cat}@k$ [15] ($aN^c$) | [0, 1.0] | The joint value of the rank of the GT POIs in $k$ POI recommendation list and the diversity of the categories retrieved by the GT POIs. | Hit only |
| | $\alpha$ - $NDCG^{geo}@k$ [15] ($aN^g$) | [0, 1.0] | The joint value of the rank of the GT POIs in $k$ POI recommendation list and the diversity of the active areas retrieved by the GT POIs. | Hit only |

*This indicates the range of target POIs to be calculated by each metric. "Hit only" uses the recommended ground-truth POIs in the recommendation list while "All" uses all POIs in the recommendation list.

5.1.2 *Evaluation Metrics.* We describe the adopted evaluation metrics in detail. A summary of the metrics is presented in Table 3. All metrics except for *EPC@k* and *AggDiv@k* are reported in the form of "per-user-average (Equation (17)),"

$$metric@k = \frac{1}{|U|} \sum_{u \in U} metric_u@k. \tag{17}$$

**Prediction accuracy:** Prediction accuracy indicates the amount of POIs in the recommendation list provided by a recommender that will actually be visited by the target user. In this article, *Precision@k* (Equation (18)) and *Recall@k* (Equation (19)) [30] are adopted. $Hit_u^k$ in Equation (18), Equation (19), and Equation (32) shows the set of "*the retrieved ground-truth POIs of u*" included in the recommendation list $R_u^k$ (*Hit POIs*, Equation (20)). We use the term *accuracy* to refer to the prediction accuracy in the later parts of this article,

$$Precison_u@k = \frac{|Hit_u^k|}{|R_u^k|}, \tag{18}$$

$$Recall_u@k = \frac{|Hit_u^k|}{|V_u^{Test}|}, \tag{19}$$

$$Hit_u^k = \left\{ v \in V \,\middle|\, v \in R_u^k \wedge v \in V_u^{Test} \right\}. \tag{20}$$

**Diversity:** Diversity indicates how many different POIs, from a certain point of view, are recommended to users. In this article, we employ *ILD@k*, *SRecall@k*, and *CPR@k* metrics for two diversification subjects: POI category and location.

*ILD@k* (Equation (21)), a variation of *ILS* [78], measures pairwise item dissimilarity in a size $k$ recommendation list. For POI categories, we use Equation (22) proposed by Castillo et al. [9] in which $dissim(i, j)$. $cat_i$ indicates the category that POI $i$ belongs to, and $sp(cat_i, cat_j)$ indicates the length of the shortest path between category $cat_i$ and $cat_j$ in the category tree. When more than one category is assigned to $i$ or $j$, we take the minimum value of $sp(cat_i, cat_j)$ from all possible combinations, and we use km distance as $dissim(i, j)$ for the POI location. The *ILD* variants are denoted as $ILD^{cat}@k$ and $ILD^{geo}@k$, respectively,

$$ILD_u^{cat(geo)}@k = \sum_{i,j \in R_u^k \wedge i \neq j} \frac{dissim_{cat(geo)}(i, j)}{|R_u^k| \cdot (|R_u^k| - 1)}, \tag{21}$$

$$dissim_{cat}(i, j) = 1 - \frac{1}{1 + sp(cat_i, cat_j)}, \tag{22}$$

$$dissim_{geo}(i, j) = kmDistance(i^{Loc}, j^{Loc}). \tag{23}$$

*SRecall* (Subtopic recall) [70] measures the sub-topic retrieval performance of IR systems for a given query. For instance, a query term “Falcon” indicates a bird species or a brand of airplane. In this case, the term Falcon has two subtopics, so an IR system that retrieves information about birds and airplanes performs better than an IR system that retrieves only one of the two meanings. In this article, we adopt two variations of *SRecall*: $SRecall^{cat}$ and $SRecall^{geo}$. We consider each user as an independent term. Subtopics related to each user are calculated based on his/her observed check-ins. $SRecall^{cat}@k$ calculates the number of distinct categories observed in $C_u^{Prof}$ that are retrieved by $R_u^k$ (Equation (24)). However, $SRecall^{geo}@k$ calculates the number of distinct visitable-areas in $A_u^{Prof}$ that are retrieved by $R_u^k$ (Equation (25)). Although *SRecall* checks for the existence of categories, it does not care about the frequency difference of each category. For instance, if category $cat_1$ appears three times and $cat_2$ appears two times in recommendation list $R_1$, and $cat_1$ appears only once and $cat_2$ appears four times in recommendation list $R_2$, then $R_1$ and $R_2$ are considered the same in terms of $SRecall^{cat}$,

$$SRecall_u^{cat}@k = \frac{|Hit_{u,cat}^k|}{|CAT_u^{Prof}|}, \tag{24}$$

$$SRecall_u^{geo}@k = \frac{|Hit_{u,geo}^k|}{|A_u^{Prof}|}, \tag{25}$$

$$Hit_{u,cat}^k = \left\{ cat \in CAT \,\middle|\, cat \in CAT_u^{Prof} \wedge (\exists v) \left[ v \in R_u^k \wedge cat \in CAT_v \right] \right\}, \tag{26}$$

$$Hit_{u,geo}^k = \left\{ a \in A \,\middle|\, a \in A_u^{Prof} \wedge (\exists v) \left[ v \in R_u^k \wedge v^{Loc} \in a \right] \right\}. \tag{27}$$

*CPR@k* (Equation (28)) is an accumulated *PR@k*, as described in Section 4.3.1. *CPR@k* is used to measure a type of distribution similarity between the distribution of an aspect of interest in user profile $C_u^{Prof}$ and the distribution of the aspect in the recommended list. The major difference between *SRecall* and *CPR* is in distribution awareness. For example, when user $u$ visits POIs in five different categories in total, a recommender that recommends only one POI from each of the five categories achieves maximum *SRecall* but cannot achieve maximum *CPR* if the distribution of recommended POI categories does not follow the distribution of $u$'s visited POI categories. We adopt two versions of *CPR*, $CPR^{cat}@k$ and $CPR^{geo}@k$, which adopt $PR^{cat}@k$ (Equation (7)) and $PR^{geo}@k$ (Equation (10)), respectively,

$$CPR_u^{cat(geo)}@k = \frac{\sum_{i=1}^{k} PR_u^{cat(geo)}@i}{k}. \tag{28}$$

**Novelty:** Novelty indicates how difficult the POIs in the recommended list are to find for users without specific initial recommendations. For novelty, we use *EPC@k* (Equation (29)) [52] as an evaluation metric. EPC increases as less popular *hit POIs* ($v \in Hit_u^k$ (Equation (20))) are located in higher positions of the recommended list. There are many possible ways of implementing EPC, and we follow Niemann et al.'s implementation [45]. $v_r$ in Equation (29) indicates the $r$th POI in recommendation list $R_u^k$, and $pop(v)$ in Equation (30) indicates the number of visits to POI $v$ made by all users,

$$EPC@k = \frac{\sum_{u\in U} \sum_{v_r \in R_u^k} \frac{rel_{u,k}(v_r)\cdot(1-pop_u(v_r))}{\log_2(r+1)}}{\sum_{u\in U} \sum_{v_r \in R_u^k} \frac{rel_{u,k}(v_r)}{\log_2(r+1)}}, \tag{29}$$

$$pop_u(v) = \frac{pop(v)}{\max_{v' \in V_u^{Cand}} pop(v')}, \tag{30}$$

$$rel_{u,k}(v) = \begin{cases} 1 & v \in Hit_u^k \\ 0 & otherwise \end{cases}. \tag{31}$$

**Catalog coverage:** Catalog coverage indicates [24, 30] the number of distinct items actually recommended for all users. We use the term *Coverage* to refer to catalog coverage. Coverage is important to business owners who register their items in a recommendation system. We further adopt aggregate diversity [21] (Equation (32)). *EPC* and *AggDiv* are calculated by using only Hit POIs, because if we consider all POIs in the recommendation list, then high *EPC* and *AggDiv* are easily achieved by recommending POIs randomly.

$$AggDiv@k = \left| \cup_{u\in U} Hit_u^k \right|. \tag{32}$$

**Joint metric:** $\alpha$-$NDCG@k$ [15] jointly measures diversity and accuracy by Equation (33). *DCG@k* has smaller value as ground-truth POIs are retrieved at lower rank (accuracy) or more retrieved information nuggets are duplicated with the information nuggets retrieved by the ground-truth POIs at higher ranks (diversity). The balance between accuracy and diversity is controlled by the value $\alpha$. We set $\alpha = 0.5$ to give the same weight to the two factors. $I_u$ in Equation (34) indicates the set of information nugget for user $u$. $I_u$ equals $A_u^{Porf}$ and $CAT_u^{Porf}$ for geographic version and category version, respectively. $Ideal\_DCG_u@k$ indicates the maximum possible value achieved by the oracle recommender,

$$\alpha\text{-}NDCG_u^{cat(geo)}@k = \frac{DCG_u^{cat(geo)}@k}{Ideal\_DCG_u^{cat(geo)}@k}, \tag{33}$$

Table 4. Base POI Recommenders

| Group | Algorithm | Personalization method | Geo-Proximity to user check-ins | Visit Info.* of friends | Category preference |
|---|---|---|---|---|---|
| State-of-the-art | USG [63] | Memory based, User based CF (UCF) | ✓ | ✓ | - |
| | GeoSoCa [72] | Content-based filtering | ✓ | ✓ | ✓ |
| | GeoBPR [67] | Model based, Learning to Rank (BPR) | ✓ | - | - |
| Vanilla (Standard) | UCF | Memory based, User-based CF (UCF) | - | - | - |
| | Ca | Content-based filtering | - | - | ✓ |
| | BPR | Model based, Learning to Rank (BPR) | - | - | - |
| Naive baseline | Rand | N/A | - | - | - |
| | Pop | N/A | - | - | - |

*Only available with Yelp dataset.

$$DCG_u^{cat(geo)}@k = \sum_{j=1}^{k} \frac{\left(\sum_{i \in I_u} J(v_j, i)(1-\alpha)^{r_{i,j-1}}\right)}{\log_2(1+j)}, \tag{34}$$

$$J(v_j, i)^{cat(geo)} = \begin{cases} 1 & cat \wedge v_j \in Hit_u^k \wedge i \in CAT_{v_j}. \\ 1 & geo \wedge v_j \in Hit_u^k \wedge v_j^{Loc} \text{ is included in } i., \\ 0 & otherwise \end{cases} \tag{35}$$

$$r_{i,k-1} = \sum_{j=1}^{k-1} J(v_j, i)^{cat(geo)}. \tag{36}$$

### 5.2 Performance Evaluation of Existing POI Recommender Algorithms

In this subsection, we evaluate the performance of existing POI recommender algorithms. Because the proposed method re-orders the POIs in the recommended list calculated by a base recommender, a performance analysis of possible base POI recommenders is important to analyze the performance of the proposed method more clearly. The reported performance also provides insight into the performance of existing POI recommendation algorithms that mostly have been evaluated using accuracy metrics only.

*5.2.1 Evaluated POI Recommender Algorithms.* We describe the algorithms selected for evaluation and the reason for their selection. We selected the following eight algorithms: three state-of-the-art algorithms, three vanilla algorithms (standard algorithms), and two naive baselines. Table 4 summarizes the eight algorithms and the parameter sets for each algorithm are shown in Table 5.

- **User-Social-Geographic (USG)** [63]**:** USG integrates (1) the visit information of friends and (2) the geographic proximity of candidate POIs to the POIs visited by the target user with (3) "memory-based" user-based collaborative filtering. The POIs previously visited by friends of the target user, or the POIs closely located to the POIs previously visited by the target user, have greater chance of appearing in the recommendation list. USG is one of the pioneer works in POI recommendation.
- **Geographic-Social-Category (GeoSoCa)** [72]: A state-of-the-art POI recommender. The method exploits (1) the geographic proximity, (2) the visit information of friends, (3) user preference to POI categories, and (4) POI popularity. GeoSoCa is based on content-based filtering [46], because the method exploits user preferences about POI categories that can be considered as descriptions of POIs.

Table 5. Tuned Parameters for Base POI Recommenders

| Group | Algorithm | Parameters[1] |
|---|---|---|
| State-of-the-art | USG_Acc [63] | $\alpha$: friend visit weight, $\beta$:geographic weight, $nn_u$: the number of nearest neighbors.<br>P, L: $\alpha = 0.2$, $\beta = 0.2$, $nn_u = 1200$.<br>N, T: $\alpha = 0.0$, $\beta = 0.2$, $nn_u = 1200$. |
| | USG_GeoDiv [63] | P, L: $\alpha = 0.2$, $\beta = 0.4$, $nn_u = 1200$.<br>N: $\alpha = 0.0$, $\beta = 0.3$, $nn_u = 1200$. T: $\alpha = 0.0$, $\beta = 0.4$, $nn_u = 1200$. |
| | GeoSoCa [72] | N/A |
| | GeoBPR [67] | $k$: the number of latent factors. $\lambda_u$, $\lambda_\pi$, $\beta_\pi$: regularization parameters,<br>$\eta$: leaning rate. $\mu$: geographic distance threshold (km).<br>P: $k = 140$, $\lambda_u = 0.03$, $\lambda_\pi = 0.03$, $\beta_\pi = 0.05$, $\eta = 0.05$, $\mu = 0.6$.<br>L: $k = 140$, $\lambda_u = 0.08$, $\lambda_\pi = 0.02$, $\beta_\pi = 0.05$, $\eta = 0.05$, $\mu = 0.3$.<br>N: $k = 160$, $\lambda_u = 0.03$, $\lambda_\pi = 0.03$, $\beta_\pi = 0.05$, $\eta = 0.05$, $\mu = 0.6$.<br>T: $k = 180$, $\lambda_u = 0.08$, $\lambda_\pi = 0.02$, $\beta_\pi = 0.05$, $\eta = 0.05$, $\mu = 0.6$. |
| Vanilla (Standard) | UCF | A: $\alpha = 0.0$, $\beta = 0.0$, $nn_u = 1200$. |
| | Ca | N/A |
| | BPR | P: $k = 110$, $\lambda_u = 0.03$, $\lambda_\pi = 0.03$, $\beta_\pi = 0.05$, $\eta = 0.05$.<br>L: $k = 110$, $\lambda_u = 0.08$, $\lambda_\pi = 0.02$, $\beta_\pi = 0.05$, $\eta = 0.05$.<br>N: $k = 150$, $\lambda_u = 0.03$, $\lambda_\pi = 0.03$, $\beta_\pi = 0.05$, $\eta = 0.05$.<br>T: $k = 170$, $\lambda_u = 0.08$, $\lambda_\pi = 0.02$, $\beta_\pi = 0.05$, $\eta = 0.05$. |
| Naive baseline | Rand | N/A |
| | Pop | N/A |

[1]P: Phoenix, L: Las Vegas, N: New York, T: Tokyo, A: All cities.

- **Geographic BPR (GeoBPR)** [67]**:** A state-of-the-art POI recommender. It is based on a "Learning to rank" model [49] to predict the pairwise item (POI) preferences of the target user. In the model training, it learns to rank methods by assuming that users prefer already visited POIs to other POIs. In addition to this basic assumption, GeoBPR assumes that, between unvisited POIs, users prefer the POIs more closely located to POIs previously visited than to POIs located remotely.
- **User-based CF (UCF):** Standard memory-based, user-based collaborative filtering algorithm. As an implementation we use USG neither considering the geographic proximity nor the visit information of friends.
- **Category (Ca):** A variant of GeoSoCa but without the geographic proximity or the visit information of friends. Ca only exploits user preference to POI categories and POI popularity.
- **Bayesian Personalized Rank (BPR)** [49]: A standard learning to rank algorithm. As an implementation, we use GeoBPR without considering geographic proximity. As a result, BPR only recognizes the user preference difference between visited POIs and other POIs.
- **Rand (Random):** A recommender that randomly recommends POIs.
- **Pop (Popularity):** A recommender that recommends the top-$k$ popular POIs.

We select USG, GeoSoCa, and GeoBPR as state-of-the-art POI recommender algorithms for examination, because the three algorithms are based on both the popular types of recommendation algorithms and the popular features exploited in POI recommendation. User-based CF adopted by USG is one of the most popular memory-based CF algorithms. BPR adopted by GeoBPR is one of the popular model-based CF algorithms that works well on ranking prediction [49, 67], i.e., predicting which item is more preferred by a given user. GeoSoCa is based on content-based filtering that is one of the popular non-collaborative filtering methods for recommendation.

Table 6. Performance of the Base POI Recommenders for Phoenix

| **Algorithm** | ***P*@5** | ***R*@5** | ***IL*$^c$@5** | ***SR*$^c$@5** | ***CP*$^c$@5** | ***aN*$^c$@5** | ***IL*$^g$@5** | ***SR*$^g$@5** | ***CP*$^g$@5** | ***aN*$^g$@5** | ***EP*@5** | ***AD*@5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rand | 0.001 | 0.003 | **0.777** | 0.088 | 0.304 | 0.001 | **10.266** | **0.663** | **0.215** | 0.002 | **0.846** | 87.2 |
| Pop | **0.019** | **0.032** | 0.578 | **0.209** | **0.562** | **0.026** | 6.259 | 0.522 | 0.190 | **0.029** | 0.196 | **141.8** |
| UCF | 0.029 | 0.046 | 0.555 | **0.227** | **0.613** | 0.040 | **6.480** | 0.609 | 0.238 | 0.048 | 0.363 | 319.4 |
| USG_Acc | **0.033** | **0.052** | 0.558 | 0.222 | 0.611 | **0.044** | 6.207 | **0.617** | 0.265 | 0.056 | 0.423 | 461.2 |
| USG_GeoDiv | **0.033** | 0.051 | **0.611** | 0.205 | 0.593 | 0.043 | 5.178 | 0.596 | **0.312** | **0.060** | **0.553** | **749.6** |
| Ca | 0.019 | 0.032 | 0.576 | **0.213** | **0.709** | 0.026 | **6.879** | **0.550** | 0.196 | 0.029 | 0.310 | 191.8 |
| GeoSoCa | **0.022** | **0.037** | **0.671** | 0.198 | 0.604 | **0.027** | 2.185 | 0.527 | **0.257** | **0.044** | **0.737** | **943.6** |
| BPR | **0.033** | **0.053** | **0.648** | 0.215 | 0.594 | 0.042 | 5.359 | 0.633 | 0.257 | **0.058** | **0.606** | 634.8 |
| GeoBPR | **0.033** | 0.051 | 0.626 | **0.219** | **0.613** | **0.043** | **5.854** | **0.653** | **0.269** | 0.056 | 0.584 | **676.8** |
| **Algorithm** | ***P*@10** | ***R*@10** | ***IL*$^c$@10** | ***SR*$^c$@10** | ***CP*$^c$@10** | ***aN*$^c$@10** | ***IL*$^g$@10** | ***SR*$^g$@10** | ***CP*$^g$@10** | ***aN*$^g$@10** | ***EP*@10** | ***AD*@10** |
| Rand | 0.001 | 0.005 | **0.777** | 0.163 | 0.330 | 0.002 | **10.265** | **0.821** | **0.287** | 0.003 | **0.846** | 174.4 |
| Pop | **0.017** | **0.055** | 0.581 | **0.308** | **0.587** | **0.034** | 6.595 | 0.641 | 0.239 | **0.039** | 0.267 | **246.6** |
| UCF | 0.025 | 0.078 | 0.575 | **0.327** | **0.633** | 0.051 | **6.798** | 0.720 | 0.294 | 0.063 | 0.420 | 531.6 |
| USG_Acc | **0.028** | **0.087** | 0.581 | 0.324 | 0.631 | **0.056** | 6.521 | **0.722** | 0.323 | 0.072 | 0.476 | 777.0 |
| USG_GeoDiv | 0.027 | 0.080 | **0.641** | 0.303 | 0.609 | 0.053 | 5.327 | 0.677 | **0.362** | **0.074** | **0.585** | **1160.6** |
| Ca | 0.016 | 0.054 | 0.599 | **0.313** | **0.731** | 0.034 | **6.859** | **0.655** | 0.248 | 0.039 | 0.361 | 316.6 |
| GeoSoCa | **0.019** | **0.062** | **0.679** | 0.308 | 0.626 | **0.035** | 2.418 | 0.569 | **0.275** | **0.057** | **0.754** | **1453.2** |
| BPR | **0.029** | **0.090** | **0.653** | 0.328 | 0.618 | **0.055** | 5.681 | 0.726 | 0.308 | **0.075** | **0.625** | 952.6 |
| GeoBPR | 0.028 | 0.087 | 0.640 | **0.330** | **0.634** | **0.055** | **6.107** | **0.749** | **0.325** | 0.074 | 0.611 | **1049.6** |
| **Algorithm** | ***P*@20** | ***R*@20** | ***IL*$^c$@20** | ***SR*$^c$@20** | ***CP*$^c$@20** | ***aN*$^c$@20** | ***IL*$^g$@20** | ***SR*$^g$@20** | ***CP*$^g$@20** | ***aN*$^g$@20** | ***EP*@20** | ***AD*@20** |
| Rand | 0.001 | 0.010 | **0.777** | 0.280 | 0.351 | 0.003 | **10.266** | **0.921** | **0.367** | 0.005 | **0.847** | 335.2 |
| Pop | **0.014** | **0.091** | 0.608 | **0.419** | **0.606** | **0.044** | 6.955 | 0.744 | 0.298 | **0.052** | 0.342 | **437.8** |
| UCF | 0.021 | 0.129 | 0.603 | **0.439** | **0.647** | 0.066 | **7.200** | **0.814** | 0.356 | 0.081 | 0.481 | 846.4 |
| USG_Acc | **0.023** | **0.142** | 0.611 | 0.437 | 0.645 | **0.071** | 6.923 | 0.813 | 0.388 | **0.092** | 0.531 | 1270.0 |
| USG_GeoDiv | 0.021 | 0.120 | **0.673** | 0.418 | 0.612 | 0.064 | 5.512 | 0.751 | **0.414** | 0.090 | **0.618** | **1757.2** |
| Ca | 0.014 | 0.094 | 0.621 | 0.412 | **0.739** | 0.045 | **7.113** | **0.761** | **0.306** | 0.052 | 0.418 | 535.6 |
| GeoSoCa | **0.016** | **0.102** | **0.687** | **0.436** | 0.641 | **0.046** | 2.700 | 0.612 | 0.297 | **0.074** | **0.770** | **2145.8** |
| BPR | **0.024** | **0.148** | **0.660** | 0.449 | 0.635 | 0.070 | 6.094 | 0.807 | 0.363 | **0.096** | **0.648** | 1380.2 |
| GeoBPR | **0.024** | 0.143 | 0.657 | **0.453** | **0.647** | **0.071** | **6.424** | **0.829** | **0.386** | 0.095 | 0.640 | **1567.0** |

From the perspective of exploited features, geo-proximity considered by the three state-of-the-art algorithms is the most popular feature (Section 2.1) in POI recommendation. Visit information of friends exploited in USG and GeoSoCa is also a popular feature. We do not select algorithms that exploit visiting time, another popular feature, because our problem is not a visiting-time-critical problem. To simplify the analysis, we also do not select algorithms that consider user comments or opinions of visited POIs.

Rand is selected to make sure that the results from the other algorithms are not randomly generated and Pop is selected as a naive baseline. UCF, Ca, and BPR are selected to examine the recommendation performance when USG, GeoSoCa, and GeoBPR do not exploit either the geographic proximity or friend visit information.

*5.2.2 Performance of POI Recommender Algorithms.* We analyze the performance of the selected algorithms from various perspectives. We first describe the performance of the baselines and then state-of-the-art algorithms with their standard counter-parts are compared to analyze the effect of adopted features such as geographic proximity and friend visit information. Finally, the effect of adopted features is summarized at the end of this subsection.

Table 6, 7, 8, and 9 show the performances of the algorithms in Phoenix, Las Vegas, New York, and Tokyo, respectively. For USG, we show the results for two different parameter tunings:

Table 7. Performance of the Base POI Recommenders for Las Vegas

| **Algorithm** | ***P*@5** | ***R*@5** | **$IL^c$@5** | **$SR^c$@5** | **$CP^c$@5** | **$aN^c$@5** | **$IL^g$@5** | **$SR^g$@5** | **$CP^g$@5** | **$aN^g$@5** | ***EP*@5** | ***AD*@5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rand | 0.002 | 0.003 | **0.778** | 0.081 | 0.278 | 0.002 | **5.528** | **0.734** | **0.272** | 0.003 | **0.823** | 42.4 |
| Pop | **0.028** | **0.057** | 0.725 | **0.169** | **0.459** | **0.033** | 3.100 | 0.581 | 0.247 | **0.052** | 0.132 | **56.4** |
| UCF | 0.047 | 0.087 | 0.665 | **0.184** | 0.582 | 0.062 | **3.582** | **0.655** | 0.295 | 0.088 | 0.320 | 128.0 |
| USG_Acc | 0.050 | **0.094** | 0.659 | 0.181 | **0.583** | **0.067** | 3.400 | **0.655** | 0.317 | 0.098 | 0.373 | 175.4 |
| USG_GeoDiv | **0.051** | 0.093 | **0.666** | 0.177 | 0.569 | **0.067** | 2.842 | 0.632 | **0.350** | **0.100** | **0.466** | **262.0** |
| Ca | **0.031** | **0.063** | 0.677 | 0.167 | **0.594** | **0.042** | **3.415** | **0.615** | 0.262 | **0.059** | 0.216 | 81.6 |
| GeoSoCa | 0.026 | 0.050 | **0.692** | **0.179** | 0.553 | 0.034 | 1.061 | 0.553 | **0.306** | 0.055 | **0.688** | **339.4** |
| BPR | 0.050 | 0.093 | 0.670 | **0.198** | 0.579 | 0.064 | **3.368** | 0.675 | 0.311 | 0.094 | **0.503** | 243.2 |
| GeoBPR | **0.051** | **0.094** | **0.671** | 0.195 | **0.594** | **0.069** | 3.268 | **0.678** | **0.334** | **0.099** | 0.498 | **279.6** |
| **Algorithm** | ***P*@10** | ***R*@10** | **$IL^c$@10** | **$SR^c$@10** | **$CP^c$@10** | **$aN^c$@10** | **$IL^g$@10** | **$SR^g$@10** | **$CP^g$@10** | **$aN^g$@10** | ***EP*@10** | ***AD*@10** |
| Rand | 0.002 | 0.008 | **0.778** | 0.150 | 0.302 | 0.003 | **5.529** | **0.864** | **0.353** | 0.005 | **0.826** | 86.6 |
| Pop | **0.024** | **0.095** | 0.711 | **0.268** | **0.489** | **0.045** | 3.562 | 0.707 | 0.300 | **0.068** | 0.218 | **99.6** |
| UCF | 0.039 | 0.140 | **0.688** | **0.291** | 0.599 | 0.078 | **3.823** | **0.760** | 0.353 | 0.112 | 0.386 | 211.6 |
| USG_Acc | **0.042** | **0.150** | 0.684 | 0.288 | **0.600** | **0.084** | 3.632 | 0.754 | 0.374 | **0.123** | 0.433 | 293.2 |
| USG_GeoDiv | 0.039 | 0.140 | **0.688** | 0.276 | 0.587 | 0.081 | 2.957 | 0.708 | **0.398** | 0.122 | **0.509** | **408.4** |
| Ca | **0.027** | **0.104** | 0.690 | 0.277 | **0.634** | **0.055** | **3.948** | **0.747** | **0.323** | **0.077** | 0.301 | 139.8 |
| GeoSoCa | 0.022 | 0.084 | **0.701** | **0.288** | 0.577 | 0.045 | 1.282 | 0.591 | 0.322 | 0.071 | **0.703** | **514.8** |
| BPR | **0.042** | **0.153** | 0.683 | **0.306** | 0.600 | 0.083 | **3.580** | **0.770** | 0.366 | 0.120 | 0.532 | 343.6 |
| GeoBPR | **0.042** | 0.150 | **0.687** | 0.303 | **0.612** | **0.086** | 3.440 | 0.766 | **0.391** | **0.124** | **0.533** | **396.0** |
| **Algorithm** | ***P*@20** | ***R*@20** | **$IL^c$@20** | **$SR^c$@20** | **$CP^c$@20** | **$aN^c$@20** | **$IL^g$@20** | **$SR^g$@20** | **$CP^g$@20** | **$aN^g$@20** | ***EP*@20** | ***AD*@20** |
| Rand | 0.002 | 0.014 | **0.778** | 0.262 | 0.321 | 0.004 | **5.538** | **0.943** | **0.437** | 0.007 | **0.827** | 167.4 |
| Pop | **0.021** | **0.157** | 0.721 | **0.392** | **0.506** | **0.060** | 4.123 | 0.825 | 0.372 | **0.090** | 0.313 | **176.6** |
| UCF | 0.031 | 0.219 | 0.706 | **0.430** | 0.606 | 0.098 | **4.150** | **0.849** | 0.417 | 0.139 | 0.451 | 334.0 |
| USG_Acc | **0.033** | **0.231** | 0.703 | 0.425 | **0.608** | **0.104** | 3.926 | 0.840 | 0.440 | **0.150** | 0.492 | 468.4 |
| USG_GeoDiv | 0.029 | 0.199 | **0.712** | 0.401 | 0.590 | 0.096 | 3.038 | 0.776 | **0.450** | 0.143 | **0.550** | **609.4** |
| Ca | **0.023** | **0.169** | 0.706 | **0.427** | **0.657** | **0.073** | **4.280** | **0.837** | **0.393** | **0.100** | 0.381 | 246.0 |
| GeoSoCa | 0.019 | 0.136 | **0.709** | 0.422 | 0.592 | 0.058 | 1.531 | 0.637 | 0.343 | 0.091 | **0.716** | **740.0** |
| BPR | **0.033** | **0.236** | 0.697 | 0.436 | 0.611 | 0.104 | **3.871** | **0.853** | 0.427 | 0.149 | 0.565 | 475.4 |
| GeoBPR | **0.033** | 0.229 | **0.702** | **0.437** | **0.620** | **0.107** | 3.664 | 0.844 | **0.452** | **0.152** | **0.573** | **583.4** |

*Recall*@20 maximized tuning (USG_Acc) and $CPR^{geo}$@20 maximized tuning (USG_GeoDiv). For the other algorithms, the tuning that achieves the maximum *Recall*@20 also achieves the maximum $CPR^{geo}$@20.

**Baseline algorithms (Rand and Pop):** Random recommendation (Rand) cannot achieve better performance than the other algorithms in accuracy metrics that require proper prediction about visited POIs (*Precision*, *Recall*, and $\alpha$-*NDCG*). One exception is the novelty metric *EPC*. Rand achieves the highest EPC, because it recommends extremely unpopular POIs visited by users by chance. In diversity metrics (*ILD*, *SRecall*, and *CPR*) and Coverage (*AggDiv*), the performance of Rand varies depends on each metric.

Top popular POI recommendation (Pop) achieves higher performance than Rand but achieves lower performance than most of the personalized recommendations in accuracy metrics. For example, compared to the personalized algorithms except Ca and GeoSoCa, Pop shows the worst performance in *Precision*, *Recall*, and $\alpha$-$NDCG^{cat/geo}$. Pop shows the lowest novelty (*EPC*) value among all algorithms, because it recommends only the most popular POIs.

**USG to UCF:** Compared to UCF, USG_Acc shows better performance in accuracy (*Precision*, *Recall*, and $\alpha$-$NDCG^{cat/geo}$), a geographic diversity ($CPR^{geo}$), novelty (*EPC*), and coverage (*AggDiv*). USG_GeoDiv further improves the beyond accuracy metrics achieved by USG_Acc in return for lower *Precision* and *Recall* compared to USG_Acc, and sometimes to "UCF".

Table 8. Performance of the Base POI Recommenders for New York

| **Algorithm** | ***P*@5** | ***R*@5** | **$IL^c$@5** | **$SR^c$@5** | **$CP^c$@5** | **$aN^c$@5** | **$IL^g$@5** | **$SR^g$@5** | **$CP^g$@5** | **$aN^g$@5** | ***EP*@5** | ***AD*@5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rand | 0.003 | 0.004 | **0.783** | **0.067** | **0.406** | 0.002 | **2.517** | **0.490** | **0.310** | 0.003 | **0.923** | 64.8 |
| Pop | **0.032** | **0.039** | 0.721 | 0.065 | 0.402 | **0.026** | 1.824 | 0.439 | 0.289 | **0.040** | 0.304 | **71.6** |
| UCF | 0.047 | 0.059 | 0.728 | 0.077 | 0.485 | 0.045 | 2.186 | 0.468 | 0.313 | 0.062 | 0.447 | 165.8 |
| USG_Acc | 0.050 | 0.062 | 0.729 | 0.079 | 0.490 | 0.047 | 2.235 | 0.482 | 0.333 | 0.067 | 0.465 | 197.0 |
| USG_GeoDiv | **0.052** | **0.064** | **0.734** | **0.081** | **0.495** | **0.048** | **2.327** | **0.492** | **0.353** | **0.071** | **0.487** | **248.6** |
| Ca | 0.026 | 0.035 | 0.635 | **0.102** | **0.682** | **0.031** | **1.998** | **0.465** | 0.307 | 0.036 | 0.512 | 142.4 |
| GeoSoCa | **0.029** | **0.036** | **0.722** | 0.100 | 0.619 | **0.031** | 0.424 | 0.347 | **0.371** | **0.050** | **0.820** | **488.0** |
| BPR | **0.056** | **0.069** | 0.725 | **0.096** | **0.592** | **0.058** | **2.410** | **0.499** | 0.361 | **0.080** | **0.744** | **495.8** |
| GeoBPR | 0.052 | 0.063 | **0.733** | **0.096** | 0.587 | 0.054 | 2.058 | 0.493 | **0.363** | 0.077 | 0.696 | 410.8 |
| **Algorithm** | ***P*@10** | ***R*@10** | **$IL^c$@10** | **$SR^c$@10** | **$CP^c$@10** | **$aN^c$@10** | **$IL^g$@10** | **$SR^g$@10** | **$CP^g$@10** | **$aN^g$@10** | ***EP*@10** | ***AD*@10** |
| Rand | 0.003 | 0.007 | **0.783** | **0.123** | 0.440 | 0.003 | **2.510** | **0.649** | **0.404** | 0.005 | **0.926** | **125.8** |
| Pop | **0.024** | **0.056** | 0.711 | 0.108 | **0.457** | **0.029** | 1.887 | 0.584 | 0.368 | **0.047** | 0.375 | 125.4 |
| UCF | 0.039 | 0.093 | 0.739 | 0.134 | 0.523 | 0.053 | 2.302 | 0.613 | 0.397 | 0.077 | 0.523 | 302.6 |
| USG_Acc | 0.042 | 0.099 | 0.742 | 0.138 | 0.530 | 0.056 | 2.372 | **0.623** | 0.425 | 0.084 | 0.545 | 388.8 |
| USG_GeoDiv | **0.043** | **0.101** | **0.748** | **0.144** | **0.537** | **0.057** | **2.450** | 0.617 | **0.448** | **0.089** | **0.568** | **515.6** |
| Ca | 0.022 | 0.055 | 0.656 | 0.159 | **0.713** | **0.037** | **2.029** | **0.597** | 0.387 | 0.044 | 0.567 | 234.0 |
| GeoSoCa | **0.025** | **0.059** | **0.735** | **0.162** | 0.644 | 0.036 | 0.501 | 0.374 | **0.389** | **0.062** | **0.843** | **809.0** |
| BPR | **0.047** | **0.110** | 0.738 | 0.161 | **0.620** | **0.068** | **2.506** | **0.630** | **0.442** | **0.098** | **0.761** | **742.0** |
| GeoBPR | 0.042 | 0.098 | **0.744** | **0.162** | 0.616 | 0.062 | 2.104 | 0.602 | 0.434 | 0.094 | 0.723 | 638.4 |
| **Algorithm** | ***P*@20** | ***R*@20** | **$IL^c$@20** | **$SR^c$@20** | **$CP^c$@20** | **$aN^c$@20** | **$IL^g$@20** | **$SR^g$@20** | **$CP^g$@20** | **$aN^g$@20** | ***EP*@20** | ***AD*@20** |
| Rand | 0.003 | 0.015 | **0.783** | **0.216** | 0.466 | 0.005 | **2.516** | **0.774** | **0.512** | 0.008 | **0.924** | **243.6** |
| Pop | **0.018** | **0.084** | 0.708 | 0.174 | **0.500** | **0.035** | 2.026 | 0.711 | 0.464 | **0.057** | 0.444 | 233.8 |
| UCF | 0.032 | 0.143 | 0.753 | 0.229 | 0.560 | 0.065 | 2.430 | **0.745** | 0.498 | 0.097 | 0.599 | 528.2 |
| USG_Acc | **0.035** | **0.156** | 0.756 | 0.237 | 0.567 | **0.070** | **2.477** | 0.738 | 0.531 | 0.109 | 0.623 | 756.0 |
| USG_GeoDiv | **0.035** | 0.151 | **0.761** | **0.244** | **0.571** | **0.070** | 2.444 | 0.706 | **0.544** | **0.111** | **0.637** | **969.2** |
| Ca | 0.018 | 0.087 | 0.673 | 0.239 | **0.732** | **0.046** | **2.081** | **0.714** | **0.479** | 0.056 | 0.620 | 396.0 |
| GeoSoCa | **0.020** | **0.093** | **0.745** | **0.251** | 0.658 | 0.044 | 0.611 | 0.405 | 0.409 | **0.078** | **0.862** | **1280.0** |
| BPR | **0.038** | **0.166** | 0.750 | 0.259 | **0.640** | **0.082** | **2.623** | **0.748** | **0.535** | **0.121** | **0.781** | **1100.2** |
| GeoBPR | 0.034 | 0.148 | **0.754** | **0.263** | 0.636 | 0.074 | 2.158 | 0.697 | 0.513 | 0.114 | 0.752 | 992.6 |

Unlike $CPR^{geo}$, the values in $SRecall^{geo}$ and $ILD^{geo}$ is not monotonically increased from UCF to UCF_GeoDiv. In most cases, the values of two metrics in USG_GeoDiv are smaller than the values in USG_Acc. Those indicate that the locations of recommended POIs are concentrated in a few frequently visited areas of target users as more weight is placed on the geographic proximity. $CPR^{geo}$ is increased, because the location distribution of recommended POIs meets the requirements of $CPR^{geo}$ to some extent by the concentration but the other metrics are decreased.

Category diversity metrics $CPR^{cat}$ and $ILD^{cat}$ show different trends by the generated LBSN datasets than the evaluated algorithms. For instance, with the Yelp dataset, $SRecall^{cat}$ and $CPR^{cat}$ roughly decrease in the order UCF to USG_Acc and USG_Acc to USG_GeoDiv while the two metrics increase with the Foursquare dataset. These indirectly indicate that USG's geographic model does not strongly influence to category diversity. $ILD^{cat}$ increased toward USG_GeoDiv, because POIs with additional categories are recommended by diversification.

The best values of $\alpha$-$NDCG$ are found in USG_Acc or USG_GeoDiv, because both accuracy and geo-diversity are improved in $\alpha$-$NDCG^{geo}$, and the accuracy increment is greater than the category diversity decrement in $\alpha$-$NDCG^{cat}$.

**GeoSoCa to Ca:** GeoSoCa cannot achieve consistent performance increase from Ca in accuracy metrics but far better performance in novelty and coverage metrics (*EPC* and *AggDiv*).

Table 9. Performance of the Base POI Recommenders for Tokyo

| **Algorithm** | $P$@5 | $R$@5 | $IL^c$@5 | $SR^c$@5 | $CP^c$@5 | $aN^c$@5 | $IL^g$@5 | $SR^g$@5 | $CP^g$@5 | $aN^g$@5 | $EP$@5 | $AD$@5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rand | 0.002 | 0.004 | **0.763** | **0.121** | **0.368** | 0.002 | 3.367 | 0.595 | 0.284 | 0.004 | **0.924** | 29.0 |
| Pop | **0.029** | **0.045** | 0.703 | 0.103 | 0.363 | **0.029** | **3.449** | **0.628** | **0.293** | **0.045** | 0.285 | **45.4** |
| UCF | 0.043 | 0.070 | 0.688 | 0.150 | 0.488 | 0.052 | **3.242** | 0.620 | 0.313 | 0.070 | 0.452 | 108.0 |
| USG_Acc | **0.046** | **0.074** | 0.685 | 0.151 | 0.493 | 0.055 | 3.170 | **0.625** | 0.338 | 0.076 | 0.473 | 121.0 |
| USG_GeoDiv | **0.046** | 0.072 | **0.699** | **0.161** | **0.507** | **0.056** | 2.934 | 0.619 | **0.401** | **0.081** | **0.518** | **177.0** |
| Ca | **0.028** | **0.046** | 0.644 | **0.190** | **0.612** | **0.040** | **3.469** | **0.608** | 0.298 | 0.045 | 0.396 | 85.2 |
| GeoSoCa | 0.027 | 0.043 | **0.672** | 0.184 | 0.602 | 0.036 | 0.454 | 0.438 | **0.384** | **0.052** | **0.758** | **233.0** |
| BPR | **0.048** | **0.078** | 0.671 | **0.165** | **0.549** | **0.060** | **2.795** | **0.607** | 0.350 | **0.082** | **0.672** | **226.2** |
| GeoBPR | 0.042 | 0.066 | **0.698** | 0.163 | 0.512 | 0.052 | 2.363 | 0.592 | **0.378** | 0.077 | 0.670 | 211.2 |
| **Algorithm** | $P$@10 | $R$@10 | $IL^c$@10 | $SR^c$@10 | $CP^c$@10 | $aN^c$@10 | $IL^g$@10 | $SR^g$@10 | $CP^g$@10 | $aN^g$@10 | $EP$@10 | $AD$@10 |
| Rand | 0.002 | 0.008 | **0.763** | **0.203** | **0.401** | 0.003 | **3.366** | **0.735** | **0.367** | 0.005 | **0.932** | 55.2 |
| Pop | **0.023** | **0.071** | 0.742 | 0.177 | 0.375 | **0.035** | 3.126 | 0.731 | 0.366 | **0.056** | 0.367 | **78.8** |
| UCF | 0.034 | 0.109 | 0.704 | 0.218 | 0.506 | 0.063 | **3.191** | 0.734 | 0.390 | 0.087 | 0.523 | 182.6 |
| USG_Acc | **0.036** | **0.113** | 0.702 | 0.224 | 0.514 | **0.066** | 3.117 | **0.737** | 0.422 | 0.094 | 0.543 | 215.4 |
| USG_GeoDiv | 0.034 | 0.106 | **0.723** | **0.250** | **0.533** | **0.066** | 2.629 | 0.692 | **0.478** | **0.097** | **0.582** | **317.8** |
| Ca | **0.023** | **0.074** | 0.661 | **0.279** | **0.654** | **0.051** | **3.367** | **0.739** | 0.383 | 0.057 | 0.475 | 151.8 |
| GeoSoCa | 0.022 | 0.069 | **0.679** | 0.267 | 0.638 | 0.045 | 0.533 | 0.459 | **0.397** | **0.065** | **0.793** | **375.2** |
| BPR | **0.038** | **0.119** | 0.692 | 0.245 | **0.577** | **0.072** | **2.937** | **0.725** | 0.420 | **0.101** | 0.697 | **336.0** |
| GeoBPR | 0.033 | 0.104 | **0.714** | **0.250** | 0.545 | 0.063 | 2.421 | 0.684 | **0.439** | 0.094 | **0.701** | 328.0 |
| **Algorithm** | $P$@20 | $R$@20 | $IL^c$@20 | $SR^c$@20 | $CP^c$@20 | $aN^c$@20 | $IL^g$@20 | $SR^g$@20 | $CP^g$@20 | $aN^g$@20 | $EP$@20 | $AD$@20 |
| Rand | 0.002 | 0.015 | **0.763** | **0.313** | **0.427** | 0.005 | **3.351** | **0.836** | **0.459** | 0.008 | **0.933** | 109.2 |
| Pop | **0.017** | **0.105** | 0.754 | 0.293 | 0.386 | **0.043** | 3.261 | 0.814 | 0.446 | **0.068** | 0.444 | **139.0** |
| UCF | 0.026 | 0.163 | 0.718 | 0.303 | 0.516 | 0.076 | **3.261** | 0.822 | 0.472 | 0.106 | 0.588 | 312.0 |
| USG_Acc | **0.028** | **0.172** | 0.721 | 0.325 | 0.529 | **0.081** | 3.129 | **0.824** | 0.513 | **0.116** | 0.610 | 397.6 |
| USG_GeoDiv | 0.025 | 0.153 | **0.744** | **0.373** | **0.548** | 0.078 | 2.301 | 0.744 | **0.542** | **0.116** | **0.639** | **565.2** |
| Ca | **0.018** | **0.114** | 0.671 | **0.380** | **0.690** | **0.063** | **3.340** | **0.830** | **0.470** | 0.071 | 0.547 | 260.8 |
| GeoSoCa | 0.017 | 0.106 | **0.693** | 0.368 | 0.660 | 0.055 | 0.614 | 0.480 | 0.411 | **0.081** | **0.820** | **582.8** |
| BPR | **0.029** | **0.176** | 0.712 | 0.350 | **0.592** | **0.087** | **3.076** | **0.824** | 0.497 | **0.121** | 0.723 | 495.8 |
| GeoBPR | 0.026 | 0.154 | **0.732** | **0.370** | 0.569 | 0.077 | 2.471 | 0.765 | **0.505** | 0.114 | **0.730** | **509.8** |

Interestingly, GeoSoCa achieves drastically small $ILD^{geo}$ compared to all other algorithms. This indicates that GeoSoCa recommends POIs located in a few concentrated areas.

A possible reason for the concentration is the existence of extremely dense locations. GeoSoCa used the Adaptive Density Kernel [72] to calculate the geographic importance of a given POI. Unfortunately, with the dataset we used, an extremely high density is assigned to some locations and it overrides the effects of user-to-POI-category preference and friend visits. The novelty and coverage improvement in GeoSoCa is caused by the geographic concentration, because the concentration prevents inclusion of popular POIs located in other areas.

Although Ca outperforms Pop in $CPR^{cat}$, $CPR^{geo}$, $EPC$, and $AggDiv$, we cannot detect their consistent advantage over Pop in the accuracy metrics. One possible explanation of this result is a lack of resolution. Unlike collaborative filtering methods, Ca only uses the user's preference to POI categories and individual POI popularity. When given POIs are in the same category, only the popularity of individual POIs can discriminate the POIs. Although Ca can overcome the data sparsity problem for areas the target user does not frequently visit by using the category preference of the user, Ca cannot use the more detailed preference information used by CF-based methods, such as the similarity of visited POIs between users.

**GeoBPR to BPR:** In most cases, BPR shows the best performance in the accuracy metrics (*Precision* and *Recall*) among all comparison algorithms. In addition, BPR shows the best performance for the vanilla algorithms in novelty and coverage. This partially shows that a learning-to-rank algorithm works well in the POI ranking prediction problem not only in prediction accuracy but also in novelty and coverage.

Contrary to our expectations, GeoBPR does not outperform BPR in our problem setting. GeoBPR does not have the performance metric that consistently shows better performance than BPR. GeoBPR assumes that the user prefers a group of POIs closely located to previously visited POIs over other POIs. Unfortunately in our problem setting, this leads to recommending POIs in frequently visited areas, where the geographic proximity constraint of GeoBPR reduces the model's freedom during learning, thus resulting in worse performance.

*5.2.3 Effect of the Adopted Geographic and Social Features on Various Evaluation Perspectives.* We discuss the effect of geographic proximity and friends visit information exploited in the evaluated POI recommender algorithms by each evaluation perspective. We exclude the results of GeoSoCa from this discussion, because GeoSoCa recommends POIs located in extremely concentrated subareas.

**Accuracy:** Although considering geographic proximity and friends visit information can increase accuracy, the improvement depends on proper method selection. For instance, despite considering the two aforementioned factors, USG achieves increased *Precision* and *Recall* compared to its vanilla counterpart, but GeoBPR does not. Accuracy and the other metrics introduce a tradeoff relationship at some point. For instance, compared to USG_Acc, USG_GeoDiv achieves increased $CPR^{geo}$ and decreased *Precision* and *Recall*.

**Novelty and Coverage:** Generally, novelty is increased, because some popular POIs recommended by vanilla algorithms are replaced by less popular POIs that satisfy the geographic and social requirements. Coverage is also increased, because the set size of newly recommended POIs (less popular POIs) is generally larger than that of the replaced POIs (Popular POIs).

**Diversity:** For category diversification, although we can detect improvement in some cities or algorithms, we cannot observe consistent improvements in $SRecall^{cat}$ and $CPR^{cat}$ to each algorithm's vanilla counterpart. The low improvement in the two metrics is not a surprise, because the distributions of POI categories are almost static over geographic areas. $ILD^{cat}$ improvement detected in most cases, because diversification introduces POIs with additional categories to the recommendation list.

For geo-diversity, except for GeoBPR in Phoenix and USG_GeoDiv in New York, $CPR^{geo}$ increases while $SRecall^{geo}$ and $ILD^{geo}$ decrease as the algorithms put more weight on geo-social features. This indicates that the recommended POIs are likely to be concentrated in a small number of frequently visited areas of the user. Because "more frequently visited" areas contribute more to $CPR^{geo}$ than "less frequently visited" areas, $CPR^{geo}$ is increased to some extent when geographic proximity consideration recommends more POIs that are located near a small number of "more frequently visited" areas. Unfortunately, the geographic proximity consideration also causes the recommended POIs to be concentrated in these areas, therefore $SRecall^{geo}$ and $ILD^{geo}$ are decreased.

**Joint metric:** $\alpha$-*NDCG* increases when an algorithm achieves a balanced improvement or tradeoff between accuracy and diversity. Because of the aforementioned characteristic, the trend of $\alpha$-*NDCG* follows the combined trends of the accuracy and diversity that are described in Accuracy and Diversity labels.

### 5.3 Performance Analysis of the Proposed Method

We evaluate the performance of our proposed method[6] in this subsection. The parameter configuration is described first, and then a discussion of the performance evaluation results is provided.

[6]The codes are available at https://drive.google.com/drive/folders/1Gpjo8TPIzisR23CSx9TEhbs2xiUpuARf?usp=sharing.

The performance is evaluated with the same metrics adopted in Section 5.2. Analysis of a successful case and an un-successful case of the proposed method is also carried out. Finally, we demonstrate the difficulty of geo-diversity improvement by adopting existing category diversification methods and discuss a possible application and a limitation of the proposed method.

*5.3.1 Configurations.* The proposed method re-orders $n$ (= $4k$) POIs recommended by a given base recommender and recommends the top-$k$ POIs as the final recommendation from the re-ordered list. We set $k = 20$, because evaluating the top 10 to 20 recommended POIs is popular in many POI recommender evaluations [35, 38, 54, 76]. $n$ is empirically chosen. Four of the eight POI recommender algorithms described in Section 5.2 are adopted as the base recommenders: USG_Acc, GeoSoCa, GeoBPR, and BPR. USG_Acc, GeoSoCa, and GeoBPR are selected to confirm how the performance of POI specialized recommenders varies when we apply the proposed method. BPR is selected, because BPR achieved the best overall performance among the base recommenders.

From Table 10 to Table 13, the performances of the proposed method with the selected base recommenders are shown. For each base recommender, we plot the performance improvement (or decrement) of the two different diversification settings from the base recommender. "GeoDiv" indicates the setting tuned for balanced performance of $Recall@20$ and $CPR^{geo}@20$, and "GeoDivAcc" indicates that the setting is tuned for maximized $Recall@20$. "$+val$" in a cell indicates that the performance is improved by the $val$ amount for the base recommender, and "$-val$" indicates the $val$ amount by which performance is degraded.

To tune "GeoDiv," we follow the policy of guaranteeing some user satisfaction, because we observe a tradeoff relationship between $Recall$ and $CPR^{geo}$ when diversification weight $\lambda$ is increased. Ziegler et al. [78] reported that more than 15% of relative $Recall$ degradation from a base recommender negatively affects user satisfaction, so we applied a more conservative satisfaction threshold. From $\lambda = 0.1$, we increase $\lambda$ by 0.1 and evaluate the performance with the tuning data. The increment stops when either of the following conditions is met: Relative degradation of $Recall@20$ of the base recommender in the tuning phase reaches 10%, or $CPR^{geo}@20$ improvement becomes saturated.

*5.3.2 Performance Analysis.* We report overall performance trends. Then the effects of the base recommenders, and diversification weights on performance are described. Tables 10, 11, 12, and 13 show the performance of the proposed method in Phoenix, Las Vegas, New York, and Tokyo, respectively.

**Overall performance:** For GeoDiv settings, as we expected, $CPR^{geo}$, $SRecall^{geo}$, and $ILD^{geo}$ are significantly improved for all base recommenders for all cities. This indicates that applying the proposed method to base recommenders effectively diversifies the locations of recommended POIs. $AggDiv$ and $EPC$ (coverage and novelty, respectively) are also improved for most base recommenders except GeoSoCa in all cities. For category diversity, although $ILD^{cat}$ improved consistently for all algorithms for all cities, no consistent increasing or decreasing pattern is found in $CPR^{cat}$ and $SRecall^{cat}$. Although the improvements achieved by GeoDiv come at the cost of $Precision$ and $Recall$, in most cases the degradation is less than 10% of the base recommender's performance.

GeoDivAcc setting focuses on $Recall@20$ maximization. The performance degradation of $Precision@20$ and $Recall@20$ is less than 3% of the base recommender's performance. It is worth noting that $Recall@20$ is slightly increased in many cases. We find that $CPR^{geo}$, $SRecall^{geo}$, and $ILD^{geo}$ are still significantly improved for all base recommenders for all cities. Although the amount is smaller, $AggDiv$ and $EPC$ in GeoDivAcc show similar increment trends as GeoDiv. Category diversity metrics, $ILD^{cat}$, $CPR^{cat}$, and $SRecall^{cat}$ also show similar trends to those shown by GeoDiv.

Table 10. Performance Improvement(+)/Decrease(−) with geoDiv and the Base Recommenders for Phoenix

| Algorithm | *P*@5 | *R*@5 | $IL^c$@5 | $SR^c$@5 | $CP^c$@5 | $aN^c$@5 | $IL^g$@5 | $SR^g$@5 | $CP^g$@5 | $aN^g$@5 | *EP*@5 | *AD*@5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BPR | **0.033** | **0.053** | 0.648 | **0.215** | **0.594** | **0.042** | 5.359 | 0.633 | 0.257 | 0.058 | 0.606 | 634.8 |
| ΔGeoDiv (0.3) | −0.003$^2$ | −0.005$^2$ | **+0.015**$^2$ | −0.012$^2$ | −0.014$^2$ | −0.004$^2$ | **+2.086**$^2$ | **+0.091**$^2$ | **+0.117**$^2$ | −0.002$^2$ | **+0.090**$^2$ | **+262.4**$^2$ |
| ΔGeoDivAcc (0.1) | **+0.000**$^1$ | −0.001$^n$ | +0.004$^1$ | −0.008$^2$ | −0.007$^2$ | **0.000**$^n$ | +1.041$^2$ | +0.051$^2$ | +0.080$^2$ | **+0.002**$^2$ | +0.055$^2$ | +170.4$^2$ |
| USG_Acc | 0.033 | 0.052 | 0.558 | **0.222** | **0.611** | **0.044** | 6.207 | 0.617 | 0.265 | 0.056 | 0.423 | 461.2 |
| ΔGeoDiv (0.3) | −0.000$^n$ | −0.001$^2$ | **+0.061**$^2$ | −0.014$^2$ | −0.014$^2$ | −0.002$^2$ | **+1.671**$^2$ | **+0.088**$^2$ | **+0.092**$^2$ | +0.002$^2$ | **+0.143**$^2$ | **+283.6**$^2$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^2$ | **+0.001**$^1$ | +0.020$^2$ | −0.006$^2$ | −0.003$^2$ | **+0.000**$^n$ | +0.542$^2$ | +0.033$^2$ | +0.042$^2$ | **+0.003**$^2$ | +0.060$^2$ | +133.0$^2$ |
| GeoSoCa | 0.022 | **0.037** | 0.671 | **0.198** | **0.604** | **0.027** | 2.185 | 0.527 | 0.257 | **0.044** | 0.737 | 943.6 |
| ΔGeoDiv (0.4) | −0.002$^2$ | −0.004$^2$ | **+0.013**$^2$ | −0.003$^2$ | −0.018$^2$ | −0.003$^2$ | **+1.737**$^2$ | **+0.060**$^2$ | **+0.070**$^2$ | −0.005$^2$ | **+0.037**$^2$ | +24.4$^n$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^n$ | **+0.000**$^n$ | +0.002$^2$ | −0.001$^2$ | −0.008$^2$ | **+0.000**$^n$ | +0.913$^2$ | +0.028$^2$ | +0.041$^2$ | **0.000**$^n$ | +0.015$^2$ | **+46.0**$^2$ |
| GeoBPR | 0.033 | **0.051** | 0.626 | **0.219** | **0.613** | **0.043** | 5.854 | 0.653 | 0.269 | 0.056 | 0.584 | 676.8 |
| ΔGeoDiv (0.3) | −0.003$^2$ | −0.006$^2$ | **+0.041**$^2$ | −0.019$^2$ | −0.031$^2$ | −0.005$^2$ | **+1.972**$^2$ | **+0.085**$^2$ | **+0.110**$^2$ | −0.002$^2$ | **+0.093**$^2$ | **+216.6**$^2$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^2$ | −0.000$^n$ | +0.019$^2$ | −0.011$^2$ | −0.015$^2$ | −0.001$^1$ | +0.799$^2$ | +0.036$^2$ | +0.065$^2$ | **+0.002**$^2$ | +0.049$^2$ | +138.4$^2$ |
| **Algorithm** | ***P*@10** | ***R*@10** | **$IL^c$@10** | **$SR^c$@10** | **$CP^c$@10** | **$aN^c$@10** | **$IL^g$@10** | **$SR^g$@10** | **$CP^g$@10** | **$aN^g$@10** | ***EP*@10** | ***AD*@10** |
| BPR | **0.029** | **0.090** | 0.653 | **0.328** | **0.618** | **0.055** | 5.681 | 0.726 | 0.308 | 0.075 | 0.625 | 952.6 |
| ΔGeoDiv (0.3) | −0.003$^2$ | −0.010$^2$ | **+0.018**$^2$ | −0.010$^2$ | −0.015$^2$ | −0.006$^2$ | **+2.082**$^2$ | **+0.101**$^2$ | **+0.148**$^2$ | −0.005$^2$ | **+0.082**$^2$ | **+341.2**$^2$ |
| ΔGeoDivAcc (0.1) | −0.001$^2$ | −0.003$^2$ | +0.009$^2$ | −0.008$^2$ | −0.008$^2$ | −0.002$^1$ | +1.374$^2$ | +0.073$^2$ | +0.107$^2$ | **+0.001**$^1$ | +0.052$^2$ | +242.8$^2$ |
| USG_Acc | 0.028 | 0.087 | 0.581 | **0.324** | **0.631** | 0.056 | 6.521 | 0.722 | 0.323 | 0.072 | 0.476 | 777.0 |
| ΔGeoDiv (0.3) | −0.001$^2$ | −0.006$^2$ | **+0.064**$^2$ | −0.011$^2$ | −0.017$^2$ | −0.004$^2$ | **+1.912**$^2$ | **+0.114**$^2$ | **+0.128**$^2$ | +0.001$^n$ | **+0.127**$^2$ | **+409.8**$^2$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^2$ | **+0.003**$^2$ | +0.021$^2$ | −0.003$^2$ | −0.004$^2$ | **+0.001**$^2$ | +0.778$^2$ | +0.048$^2$ | +0.057$^2$ | **+0.004**$^2$ | +0.050$^2$ | +190.4$^2$ |
| GeoSoCa | 0.019 | 0.062 | 0.679 | 0.308 | **0.626** | 0.035 | 2.418 | 0.569 | 0.275 | **0.057** | 0.754 | 1453.2 |
| ΔGeoDiv (0.4) | −0.001$^2$ | −0.004$^2$ | **+0.009**$^2$ | +0.000$^n$ | −0.014$^2$ | −0.003$^2$ | **+1.674**$^2$ | **+0.072**$^2$ | **+0.090**$^2$ | −0.006$^2$ | **+0.026**$^2$ | +37.8$^1$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^n$ | **+0.001**$^n$ | +0.002$^2$ | **+0.001**$^1$ | −0.005$^2$ | **+0.001**$^n$ | +0.932$^2$ | +0.039$^2$ | +0.054$^2$ | **+0.000**$^n$ | +0.008$^2$ | **+43.6**$^1$ |
| GeoBPR | **0.028** | **0.087** | 0.640 | **0.330** | **0.634** | **0.055** | 6.107 | 0.749 | 0.325 | **0.074** | 0.611 | 1049.6 |
| ΔGeoDiv (0.3) | −0.003$^2$ | −0.014$^2$ | **+0.039**$^2$ | −0.018$^2$ | −0.032$^2$ | −0.008$^2$ | **+2.063**$^2$ | **+0.098**$^2$ | **+0.146**$^2$ | −0.006$^2$ | **+0.083**$^2$ | **+278.2**$^2$ |
| ΔGeoDivAcc (0.1) | −0.001$^2$ | −0.005$^2$ | +0.020$^2$ | −0.010$^2$ | −0.016$^2$ | −0.003$^2$ | +1.209$^2$ | +0.061$^2$ | +0.091$^2$ | **+0.000**$^n$ | +0.044$^2$ | +188.2$^2$ |
| **Algorithm** | ***P*@20** | ***R*@20** | **$IL^c$@20** | **$SR^c$@20** | **$CP^c$@20** | **$aN^c$@20** | **$IL^g$@20** | **$SR^g$@20** | **$CP^g$@20** | **$aN^g$@20** | ***EP*@20** | ***AD*@20** |
| BPR | **0.024** | **0.148** | 0.660 | **0.449** | **0.635** | **0.070** | 6.094 | 0.807 | 0.363 | 0.096 | 0.648 | 1380.2 |
| ΔGeoDiv (0.3) | −0.002$^2$ | −0.009$^2$ | **+0.013**$^2$ | −0.003$^2$ | −0.014$^2$ | −0.007$^2$ | **+1.498**$^2$ | **+0.068**$^2$ | **+0.153**$^2$ | −0.004$^2$ | **+0.062**$^2$ | **+385.2**$^2$ |
| ΔGeoDivAcc (0.1) | −0.000$^2$ | −0.001$^n$ | +0.008$^2$ | −0.002$^1$ | −0.008$^2$ | −0.002$^2$ | +1.274$^2$ | +0.064$^2$ | +0.124$^2$ | **+0.002**$^2$ | +0.040$^2$ | +308.8$^2$ |
| USG_Acc | 0.023 | 0.142 | 0.611 | 0.437 | **0.645** | 0.071 | 6.923 | 0.813 | 0.388 | 0.092 | 0.531 | 1270.0 |
| ΔGeoDiv (0.3) | −0.001$^2$ | −0.007$^2$ | **+0.042**$^2$ | **+0.002**$^2$ | −0.019$^2$ | −0.005$^2$ | **+1.501**$^2$ | **+0.081**$^2$ | **+0.145**$^2$ | 0.000$^n$ | **+0.091**$^2$ | **+435.4**$^2$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^2$ | **+0.002**$^1$ | +0.016$^2$ | **+0.002**$^2$ | −0.006$^2$ | **+0.001**$^1$ | +0.901$^2$ | +0.056$^2$ | +0.073$^2$ | **+0.004**$^2$ | +0.036$^2$ | +231.2$^2$ |
| GeoSoCa | 0.016 | 0.102 | 0.687 | 0.436 | **0.641** | 0.046 | 2.700 | 0.612 | 0.297 | 0.074 | 0.770 | 2145.8 |
| ΔGeoDiv (0.4) | −0.000$^2$ | −0.001$^n$ | **+0.004**$^2$ | **+0.003**$^2$ | −0.007$^2$ | −0.002$^2$ | **+1.180**$^2$ | **+0.061**$^2$ | **+0.094**$^2$ | −0.004$^2$ | **+0.014**$^2$ | **+47.2**$^2$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^2$ | **+0.003**$^2$ | −0.000$^n$ | +0.002$^2$ | −0.001$^2$ | **+0.001**$^2$ | +0.637$^2$ | +0.034$^2$ | +0.057$^2$ | **+0.001**$^2$ | +0.002$^1$ | +40.4$^2$ |
| GeoBPR | **0.024** | **0.143** | 0.657 | **0.453** | **0.647** | **0.071** | 6.424 | 0.829 | 0.386 | **0.095** | 0.640 | 1567.0 |
| ΔGeoDiv (0.3) | −0.002$^2$ | −0.015$^2$ | **+0.026**$^2$ | −0.006$^2$ | −0.030$^2$ | −0.009$^2$ | **+1.530**$^2$ | **+0.063**$^2$ | **+0.154**$^2$ | −0.007$^2$ | **+0.062**$^2$ | **+308.0**$^2$ |
| ΔGeoDivAcc (0.1) | −0.000$^2$ | −0.005$^2$ | +0.016$^2$ | −0.003$^2$ | −0.016$^2$ | −0.003$^2$ | +1.262$^2$ | +0.059$^2$ | +0.112$^2$ | −0.000$^n$ | +0.034$^2$ | +237.8$^2$ |

$^{n,\ 1,\ 2}$The gain over (or loss under) the base recommender is statistically not significant (n), significant at $p < 0.05$ (1), or significant at $p < 0.01$ (2).
*Two-tailed paired t test* was used for the statistical significance test.
The value in () in Alg. column indicates the adopted diversification weight $\lambda$.

From the results, we can see that applying the proposed method to the base recommenders improves geo-diversity (0.06 to 0.18 in $CPR^{geo}$@20), novelty, and coverage with tolerable prediction accuracy loss (0.024 loss to 0.005 increment in *Recall*@20). Even when users place more weight on accuracy, the proposed method with a small weight on diversification can improve geo-diversity (0.05 to 0.12 in $CPR^{geo}$@20), novelty, and coverage with negligible loss of accuracy (0.005 loss to 0.007 increment in *Recall*@20). Unfortunately, for category diversity, we cannot observe a consistent improvement caused by the proposed method.

Table 11. Performance Improvement(+)/Decrease(−) with geoDiv and the Base Recommenders for Las Vegas

| **Algorithm** | $P$@5 | $R$@5 | $IL^c$@5 | $SR^c$@5 | $CP^c$@5 | $aN^c$@5 | $IL^g$@5 | $SR^g$@5 | $CP^g$@5 | $aN^g$@5 | $EP$@5 | $AD$@5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BPR | 0.050 | **0.093** | 0.670 | **0.198** | **0.579** | 0.064 | 3.368 | 0.675 | 0.311 | 0.094 | 0.503 | 243.2 |
| ΔGeoDiv (0.3) | −0.004$^{2}$ | −0.009$^{2}$ | **+0.021**$^{2}$ | −0.007$^{2}$ | −0.021$^{2}$ | −0.006$^{2}$ | **+1.240**$^{2}$ | **+0.090**$^{2}$ | **+0.105**$^{2}$ | −0.004$^{2}$ | **+0.081**$^{2}$ | **+62.8**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{n}$ | **+0.000**$^{n}$ | +0.007$^{1}$ | −0.003$^{2}$ | −0.009$^{2}$ | **+0.001**$^{n}$ | +0.552$^{2}$ | +0.043$^{2}$ | +0.059$^{2}$ | **+0.003**$^{n}$ | +0.038$^{2}$ | +34.0$^{2}$ |
| USG_Acc | 0.050 | 0.094 | 0.659 | 0.181 | **0.583** | 0.067 | 3.400 | 0.655 | 0.317 | 0.098 | 0.373 | 175.4 |
| ΔGeoDiv (0.4) | −0.001$^{2}$ | −0.005$^{2}$ | **+0.033**$^{2}$ | **+0.002**$^{1}$ | −0.019$^{2}$ | −0.003$^{2}$ | **+1.236**$^{2}$ | **+0.098**$^{2}$ | **+0.091**$^{2}$ | −0.002$^{1}$ | **+0.107**$^{2}$ | **+101.0**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.002**$^{1}$ | +0.009$^{2}$ | +0.000$^{n}$ | −0.006$^{2}$ | **+0.001**$^{n}$ | +0.329$^{2}$ | +0.026$^{2}$ | +0.027$^{2}$ | **+0.002**$^{2}$ | +0.031$^{2}$ | +33.4$^{2}$ |
| GeoSoCa | **0.026** | 0.050 | 0.692 | 0.179 | 0.553 | **0.034** | 1.061 | 0.553 | 0.306 | 0.055 | 0.688 | 339.4 |
| ΔGeoDiv (0.4) | −0.002$^{1}$ | −0.003$^{1}$ | **+0.013**$^{2}$ | 0.000$^{n}$ | −0.003$^{1}$ | −0.003$^{2}$ | **+1.297**$^{2}$ | **+0.079**$^{2}$ | **+0.067**$^{2}$ | −0.004$^{2}$ | **+0.031**$^{2}$ | +13.8$^{n}$ |
| ΔGeoDivAcc (0.1) | **+0.000**$^{n}$ | **+0.001**$^{n}$ | +0.006$^{2}$ | **+0.001**$^{1}$ | **+0.004**$^{2}$ | **0.000**$^{n}$ | +0.585$^{2}$ | +0.032$^{2}$ | +0.034$^{2}$ | **+0.001**$^{n}$ | +0.009$^{2}$ | **+17.6**$^{1}$ |
| GeoBPR | **0.051** | **0.094** | 0.671 | **0.195** | **0.594** | **0.069** | 3.268 | 0.678 | 0.334 | **0.099** | 0.498 | 279.6 |
| ΔGeoDiv (0.3) | −0.005$^{2}$ | −0.010$^{2}$ | **+0.024**$^{2}$ | −0.004$^{2}$ | −0.021$^{2}$ | −0.007$^{2}$ | **+1.289**$^{2}$ | **+0.092**$^{2}$ | **+0.085**$^{2}$ | −0.008$^{2}$ | **+0.064**$^{2}$ | **+40.2**$^{2}$ |
| ΔGeoDivAcc (0.1) | −0.001$^{n}$ | −0.002$^{1}$ | +0.008$^{2}$ | −0.002$^{2}$ | −0.008$^{2}$ | −0.001$^{1}$ | +0.591$^{2}$ | +0.043$^{2}$ | +0.041$^{2}$ | −0.000$^{n}$ | +0.021$^{2}$ | +14.0$^{2}$ |
| **Algorithm** | $P$@10 | $R$@10 | $IL^c$@10 | $SR^c$@10 | $CP^c$@10 | $aN^c$@10 | $IL^g$@10 | $SR^g$@10 | $CP^g$@10 | $aN^g$@10 | $EP$@10 | $AD$@10 |
| BPR | **0.042** | **0.153** | 0.683 | **0.306** | **0.600** | **0.083** | 3.580 | 0.770 | 0.366 | 0.120 | 0.532 | 343.6 |
| ΔGeoDiv (0.3) | −0.005$^{2}$ | −0.019$^{2}$ | **+0.021**$^{2}$ | −0.001$^{n}$ | −0.020$^{2}$ | −0.009$^{2}$ | **+1.338**$^{2}$ | **+0.113**$^{2}$ | **+0.140**$^{2}$ | −0.009$^{2}$ | **+0.071**$^{2}$ | **+92.6**$^{2}$ |
| ΔGeoDivAcc (0.1) | −0.001$^{2}$ | −0.004$^{1}$ | +0.008$^{2}$ | **+0.000**$^{n}$ | −0.007$^{2}$ | −0.001$^{n}$ | +0.746$^{2}$ | +0.065$^{2}$ | +0.079$^{2}$ | **+0.001**$^{n}$ | +0.032$^{2}$ | +44.8$^{2}$ |
| USG_Acc | 0.042 | 0.150 | 0.684 | 0.288 | **0.600** | 0.084 | 3.632 | 0.754 | 0.374 | 0.123 | 0.433 | 293.2 |
| ΔGeoDiv (0.4) | −0.002$^{2}$ | −0.010$^{2}$ | **+0.027**$^{2}$ | **+0.005**$^{2}$ | −0.019$^{2}$ | −0.005$^{2}$ | **+1.314**$^{2}$ | **+0.121**$^{2}$ | **+0.130**$^{2}$ | −0.005$^{2}$ | **+0.089**$^{2}$ | **+137.2**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.003**$^{1}$ | +0.008$^{2}$ | +0.002$^{1}$ | −0.005$^{2}$ | **+0.001**$^{1}$ | +0.529$^{2}$ | +0.047$^{2}$ | +0.042$^{2}$ | **+0.002**$^{2}$ | +0.024$^{2}$ | +50.8$^{2}$ |
| GeoSoCa | **0.022** | 0.084 | 0.701 | 0.288 | 0.577 | **0.045** | 1.282 | 0.591 | 0.322 | 0.071 | 0.703 | 514.8 |
| ΔGeoDiv (0.4) | −0.001$^{2}$ | −0.004$^{1}$ | **+0.014**$^{2}$ | **+0.003**$^{2}$ | −0.004$^{2}$ | −0.004$^{2}$ | **+1.360**$^{2}$ | **+0.106**$^{2}$ | **+0.094**$^{2}$ | −0.004$^{2}$ | **+0.021**$^{2}$ | **+16.4**$^{n}$ |
| ΔGeoDivAcc (0.1) | **+0.000**$^{n}$ | **+0.001**$^{n}$ | +0.007$^{2}$ | +0.002$^{1}$ | **+0.002**$^{2}$ | **0.000**$^{n}$ | +0.664$^{2}$ | +0.049$^{2}$ | +0.047$^{2}$ | **+0.001**$^{n}$ | +0.004$^{1}$ | +15.0$^{1}$ |
| GeoBPR | **0.042** | **0.150** | 0.687 | **0.303** | **0.612** | **0.086** | 3.440 | 0.766 | 0.391 | **0.124** | 0.533 | 396.0 |
| ΔGeoDiv (0.3) | −0.004$^{2}$ | −0.017$^{2}$ | **+0.023**$^{2}$ | −0.003$^{2}$ | −0.022$^{2}$ | −0.010$^{2}$ | **+1.444**$^{2}$ | **+0.121**$^{2}$ | **+0.123**$^{2}$ | −0.012$^{2}$ | **+0.055**$^{2}$ | **+77.2**$^{2}$ |
| ΔGeoDivAcc (0.1) | −0.001$^{n}$ | −0.004$^{1}$ | +0.010$^{2}$ | −0.001$^{n}$ | −0.008$^{2}$ | −0.002$^{1}$ | +0.821$^{2}$ | +0.070$^{2}$ | +0.062$^{2}$ | −0.001$^{1}$ | +0.019$^{2}$ | +36.8$^{2}$ |
| **Algorithm** | $P$@20 | $R$@20 | $IL^c$@20 | $SR^c$@20 | $CP^c$@20 | $aN^c$@20 | $IL^g$@20 | $SR^g$@20 | $CP^g$@20 | $aN^g$@20 | $EP$@20 | $AD$@20 |
| BPR | **0.033** | **0.236** | 0.697 | 0.436 | **0.611** | **0.104** | 3.871 | 0.853 | 0.427 | 0.149 | 0.565 | 475.4 |
| ΔGeoDiv (0.3) | −0.003$^{2}$ | −0.024$^{2}$ | **+0.015**$^{2}$ | **+0.005**$^{2}$ | −0.019$^{2}$ | −0.011$^{2}$ | **+1.150**$^{2}$ | **+0.087**$^{2}$ | **+0.154**$^{2}$ | −0.011$^{2}$ | **+0.055**$^{2}$ | **+110.8**$^{2}$ |
| ΔGeoDivAcc (0.1) | −0.000$^{1}$ | −0.003$^{2}$ | +0.007$^{2}$ | +0.004$^{2}$ | −0.007$^{2}$ | −0.001$^{n}$ | +0.814$^{2}$ | +0.068$^{2}$ | +0.097$^{2}$ | **+0.001**$^{n}$ | +0.024$^{2}$ | +62.4$^{2}$ |
| USG_Acc | 0.033 | 0.231 | 0.703 | 0.425 | **0.608** | 0.104 | 3.926 | 0.840 | 0.440 | 0.150 | 0.492 | 468.4 |
| ΔGeoDiv (0.4) | −0.002$^{2}$ | −0.012$^{2}$ | **+0.017**$^{2}$ | **+0.008**$^{2}$ | −0.019$^{2}$ | −0.005$^{2}$ | **+1.034**$^{2}$ | **+0.086**$^{2}$ | **+0.146**$^{2}$ | −0.006$^{2}$ | **+0.062**$^{2}$ | **+139.4**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.004**$^{n}$ | +0.006$^{2}$ | +0.005$^{2}$ | −0.004$^{2}$ | **+0.002**$^{2}$ | +0.598$^{2}$ | +0.054$^{2}$ | +0.059$^{2}$ | **+0.003**$^{2}$ | +0.018$^{2}$ | +55.0$^{2}$ |
| GeoSoCa | 0.019 | 0.136 | 0.709 | 0.422 | 0.592 | 0.058 | 1.531 | 0.637 | 0.343 | 0.091 | 0.716 | 740.0 |
| ΔGeoDiv (0.4) | −0.000$^{n}$ | −0.000$^{n}$ | **+0.009**$^{2}$ | **+0.009**$^{2}$ | −0.003$^{2}$ | −0.003$^{2}$ | **+1.097**$^{2}$ | **+0.098**$^{2}$ | **+0.108**$^{2}$ | −0.003$^{2}$ | **+0.016**$^{2}$ | **+18.4**$^{n}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.007**$^{2}$ | +0.003$^{2}$ | +0.005$^{2}$ | **+0.003**$^{2}$ | **+0.002**$^{2}$ | +0.486$^{2}$ | +0.045$^{2}$ | +0.054$^{2}$ | **+0.003**$^{2}$ | +0.002$^{1}$ | +18.2$^{2}$ |
| GeoBPR | **0.033** | **0.229** | 0.702 | 0.437 | **0.620** | **0.107** | 3.664 | 0.844 | 0.452 | **0.152** | 0.573 | 583.4 |
| ΔGeoDiv (0.3) | −0.003$^{2}$ | −0.021$^{2}$ | **+0.016**$^{2}$ | +0.001$^{1}$ | −0.023$^{2}$ | −0.011$^{2}$ | **+1.203**$^{2}$ | **+0.090**$^{2}$ | **+0.141**$^{2}$ | −0.013$^{2}$ | **+0.038**$^{2}$ | **+80.4**$^{2}$ |
| ΔGeoDivAcc (0.1) | −0.000$^{2}$ | −0.005$^{2}$ | +0.009$^{2}$ | **+0.002**$^{2}$ | −0.009$^{2}$ | −0.002$^{2}$ | +0.921$^{2}$ | +0.076$^{2}$ | +0.086$^{2}$ | −0.002$^{1}$ | +0.013$^{2}$ | +47.8$^{2}$ |

$^{n,\ 1,\ 2}$The gain over (or loss under) the base recommender is statistically not significant (n), significant at $p < 0.05$ (1), or significant at $p < 0.01$ (2).
*Two-tailed paired t test* was used for the statistical significance test.
The value in () in Alg. column indicates the adopted diversification weight $\lambda$.

In most cases, $\alpha$-$NDGC^{geo}$ is improved by GeoDivAcc but decreased by GeoDiv. $\alpha$-$NDGC^{geo}$ considers both of ground-truth POI retrieval (similarly to *Recall*) performance and active area retrieval (similarly to $SRecall^{geo}$) performance by the "retrieved" ground-truth POIs. Due to the metric's characteristic, $\alpha$-$NDGC^{geo}$ value achieved by GeoDiv is decreased if the number of retrieved ground-truth POIs is decreased and the number of active areas retrieved by the retrieved ground truth is relatively small to the number of active areas appeared in the recommendation list. $\alpha$-$NDGC^{cat}$ is increased by neither GeoDivAcc nor GeoDiv, because the proposed method causes

Table 12. Performance Improvement(+)/Decrease(−) with geoDiv and the Base Recommenders for New York

| **Algorithm** | ***P*@5** | ***R*@5** | ***IL*$^c$@5** | ***SR*$^c$@5** | ***CP*$^c$@5** | ***aN*$^c$@5** | ***IL*$^g$@5** | ***SR*$^g$@5** | ***CP*$^g$@5** | ***aN*$^g$@5** | ***EP*@5** | ***AD*@5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BPR | 0.056 | 0.069 | 0.725 | 0.096 | 0.592 | 0.058 | 2.410 | 0.499 | 0.361 | 0.080 | 0.744 | 495.8 |
| ΔGeoDiv (0.3) | −0.001$^{n}$ | −0.001$^{n}$ | **+0.019**$^{2}$ | **+0.003**$^{2}$ | +0.000$^{n}$ | −0.002$^{2}$ | **+0.713**$^{2}$ | **+0.096**$^{2}$ | **+0.099**$^{2}$ | +0.004$^{2}$ | **+0.043**$^{2}$ | **+115.2**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{2}$ | **+0.002**$^{2}$ | +0.005$^{2}$ | +0.001$^{2}$ | **+0.001**$^{n}$ | **+0.001**$^{2}$ | +0.358$^{2}$ | +0.057$^{2}$ | +0.048$^{2}$ | **+0.005**$^{2}$ | +0.012$^{2}$ | +44.0$^{2}$ |
| USG_Acc | 0.050 | 0.062 | 0.729 | 0.079 | 0.490 | 0.047 | 2.235 | 0.482 | 0.333 | 0.067 | 0.465 | 197.0 |
| ΔGeoDiv (0.5) | +0.001$^{n}$ | +0.001$^{n}$ | **+0.030**$^{2}$ | **+0.011**$^{2}$ | **+0.029**$^{2}$ | +0.001$^{n}$ | **+1.064**$^{2}$ | **+0.106**$^{2}$ | **+0.126**$^{2}$ | **+0.008**$^{2}$ | **+0.119**$^{2}$ | **+243.6**$^{2}$ |
| ΔGeoDivAcc (0.2) | **+0.003**$^{2}$ | **+0.003**$^{2}$ | +0.008$^{2}$ | +0.003$^{2}$ | +0.007$^{2}$ | **+0.002**$^{2}$ | +0.427$^{2}$ | +0.058$^{2}$ | +0.046$^{2}$ | +0.006$^{2}$ | +0.036$^{2}$ | +74.8$^{2}$ |
| GeoSoCa | 0.029 | 0.036 | 0.722 | 0.100 | **0.619** | 0.031 | 0.424 | 0.347 | 0.371 | 0.050 | 0.820 | 488.0 |
| ΔGeoDiv (0.2) | +0.001$^{2}$ | +0.002$^{2}$ | **+0.014**$^{2}$ | **+0.001**$^{1}$ | −0.010$^{2}$ | +0.001$^{1}$ | **+0.465**$^{2}$ | **+0.059**$^{2}$ | **+0.051**$^{2}$ | +0.002$^{1}$ | **+0.023**$^{2}$ | **+32.8**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{2}$ | **+0.004**$^{2}$ | +0.006$^{2}$ | **+0.001**$^{2}$ | −0.001$^{n}$ | **+0.002**$^{2}$ | +0.332$^{2}$ | +0.045$^{2}$ | +0.037$^{2}$ | **+0.003**$^{2}$ | +0.007$^{n}$ | +31.8$^{2}$ |
| GeoBPR | 0.052 | 0.063 | 0.733 | 0.096 | **0.587** | **0.054** | 2.058 | 0.493 | 0.363 | 0.077 | 0.696 | 410.8 |
| ΔGeoDiv (0.4) | −0.003$^{2}$ | −0.006$^{2}$ | **+0.022**$^{2}$ | **+0.003**$^{2}$ | −0.005$^{1}$ | −0.005$^{2}$ | **+0.900**$^{2}$ | **+0.105**$^{2}$ | **+0.107**$^{2}$ | −0.002$^{1}$ | **+0.071**$^{2}$ | **+146.4**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.002**$^{1}$ | +0.005$^{2}$ | +0.001$^{1}$ | −0.001$^{n}$ | **+0.000**$^{n}$ | +0.404$^{2}$ | +0.055$^{2}$ | +0.043$^{2}$ | **+0.003**$^{2}$ | +0.016$^{2}$ | +52.8$^{2}$ |
| **Algorithm** | ***P*@10** | ***R*@10** | ***IL*$^c$@10** | ***SR*$^c$@10** | ***CP*$^c$@10** | ***aN*$^c$@10** | ***IL*$^g$@10** | ***SR*$^g$@10** | ***CP*$^g$@10** | ***aN*$^g$@10** | ***EP*@10** | ***AD*@10** |
| BPR | 0.047 | 0.110 | 0.738 | 0.161 | 0.620 | 0.068 | 2.506 | 0.630 | 0.442 | 0.098 | 0.761 | 742.0 |
| ΔGeoDiv (0.3) | −0.001$^{2}$ | −0.002$^{2}$ | **+0.016**$^{2}$ | **+0.007**$^{2}$ | −0.001$^{n}$ | −0.002$^{2}$ | **+0.811**$^{2}$ | **+0.135**$^{2}$ | **+0.129**$^{2}$ | +0.003$^{2}$ | **+0.035**$^{2}$ | **+154.8**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.003**$^{2}$ | +0.006$^{2}$ | +0.003$^{2}$ | **+0.001**$^{2}$ | **+0.001**$^{2}$ | +0.451$^{2}$ | +0.082$^{2}$ | +0.067$^{2}$ | **+0.005**$^{2}$ | +0.010$^{2}$ | +66.8$^{2}$ |
| USG_Acc | 0.042 | 0.099 | 0.742 | 0.138 | 0.530 | 0.056 | 2.372 | 0.623 | 0.425 | 0.084 | 0.545 | 388.8 |
| ΔGeoDiv (0.5) | +0.002$^{2}$ | +0.000$^{n}$ | **+0.021**$^{2}$ | **+0.018**$^{2}$ | **+0.028**$^{2}$ | +0.002$^{1}$ | **+0.947**$^{2}$ | **+0.130**$^{2}$ | **+0.151**$^{2}$ | **+0.007**$^{2}$ | **+0.091**$^{2}$ | **+306.4**$^{2}$ |
| ΔGeoDivAcc (0.2) | **+0.003**$^{2}$ | **+0.005**$^{2}$ | +0.008$^{2}$ | +0.008$^{2}$ | +0.010$^{2}$ | **+0.003**$^{2}$ | +0.574$^{2}$ | +0.082$^{2}$ | +0.071$^{2}$ | **+0.007**$^{2}$ | +0.034$^{2}$ | +143.4$^{2}$ |
| GeoSoCa | 0.025 | 0.059 | 0.735 | 0.162 | 0.644 | 0.036 | 0.501 | 0.374 | 0.389 | 0.062 | 0.843 | 809.0 |
| ΔGeoDiv (0.2) | +0.000$^{n}$ | +0.002$^{1}$ | **+0.007**$^{2}$ | **+0.003**$^{2}$ | −0.004$^{2}$ | +0.001$^{n}$ | **+0.505**$^{2}$ | **+0.076**$^{2}$ | **+0.069**$^{2}$ | +0.001$^{n}$ | **+0.010**$^{1}$ | +4.4$^{n}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{1}$ | **+0.005**$^{2}$ | +0.002$^{2}$ | **+0.003**$^{2}$ | **+0.003**$^{1}$ | **+0.003**$^{2}$ | +0.372$^{2}$ | +0.062$^{2}$ | +0.054$^{2}$ | **+0.004**$^{2}$ | −0.000$^{n}$ | **+14.6**$^{n}$ |
| GeoBPR | 0.042 | 0.098 | 0.744 | 0.162 | **0.616** | 0.062 | 2.104 | 0.602 | 0.434 | 0.094 | 0.723 | 638.4 |
| ΔGeoDiv (0.4) | −0.002$^{2}$ | −0.006$^{2}$ | **+0.017**$^{2}$ | **+0.007**$^{2}$ | −0.005$^{2}$ | −0.005$^{2}$ | **+0.975**$^{2}$ | **+0.151**$^{2}$ | **+0.141**$^{2}$ | −0.003$^{2}$ | **+0.056**$^{2}$ | **+194.2**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{2}$ | **+0.003**$^{1}$ | +0.006$^{2}$ | +0.003$^{2}$ | −0.001$^{n}$ | **+0.001**$^{1}$ | +0.537$^{2}$ | +0.087$^{2}$ | +0.065$^{2}$ | **+0.004**$^{2}$ | +0.014$^{2}$ | +80.2$^{2}$ |
| **Algorithm** | ***P*@20** | ***R*@20** | ***IL*$^c$@20** | ***SR*$^c$@20** | ***CP*$^c$@20** | ***aN*$^c$@20** | ***IL*$^g$@20** | ***SR*$^g$@20** | ***CP*$^g$@20** | ***aN*$^g$@20** | ***EP*@20** | ***AD*@20** |
| BPR | 0.038 | 0.166 | 0.750 | 0.259 | 0.640 | 0.082 | 2.623 | 0.748 | 0.535 | 0.121 | 0.781 | 1100.2 |
| ΔGeoDiv (0.3) | −0.001$^{2}$ | −0.004$^{2}$ | **+0.011**$^{2}$ | **+0.011**$^{2}$ | −0.002$^{2}$ | −0.003$^{2}$ | **+0.759**$^{2}$ | **+0.134**$^{2}$ | **+0.149**$^{2}$ | +0.001$^{2}$ | **+0.024**$^{2}$ | **+157.2**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.003**$^{2}$ | +0.005$^{2}$ | +0.006$^{2}$ | **+0.001**$^{2}$ | **+0.001**$^{2}$ | +0.475$^{2}$ | +0.093$^{2}$ | +0.086$^{2}$ | **+0.005**$^{2}$ | +0.007$^{2}$ | +90.2 $^{2}$ |
| USG_Acc | 0.035 | 0.156 | 0.756 | 0.237 | 0.567 | 0.070 | 2.477 | 0.738 | 0.531 | 0.109 | 0.623 | 756.0 |
| ΔGeoDiv (0.5) | +0.001$^{1}$ | −0.000$^{n}$ | **+0.010**$^{2}$ | **+0.017**$^{2}$ | **+0.019**$^{2}$ | +0.001$^{n}$ | **+0.750**$^{2}$ | **+0.116**$^{2}$ | **+0.156**$^{2}$ | +0.005$^{2}$ | **+0.059**$^{2}$ | **+271.4**$^{2}$ |
| ΔGeoDivAcc (0.2) | **+0.002**$^{2}$ | **+0.006**$^{2}$ | +0.005$^{2}$ | +0.009$^{2}$ | +0.008$^{2}$ | **+0.003**$^{2}$ | +0.602$^{2}$ | +0.094$^{2}$ | +0.094$^{2}$ | **+0.006**$^{2}$ | +0.024$^{2}$ | +166.8$^{2}$ |
| GeoSoCa | 0.020 | 0.093 | 0.745 | 0.251 | 0.658 | 0.044 | 0.611 | 0.405 | 0.409 | 0.078 | 0.862 | **1280.0** |
| ΔGeoDiv (0.2) | 0.000$^{n}$ | +0.003$^{2}$ | **+0.002**$^{2}$ | **+0.004**$^{2}$ | +0.002$^{2}$ | +0.001$^{1}$ | **+0.393**$^{2}$ | **+0.065**$^{2}$ | **+0.073**$^{2}$ | +0.001$^{n}$ | **+0.003**$^{n}$ | −36.8$^{1}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.006**$^{2}$ | −0.001$^{1}$ | +0.003$^{2}$ | **+0.007**$^{2}$ | **+0.003**$^{2}$ | +0.288$^{2}$ | +0.055$^{2}$ | +0.060$^{2}$ | **+0.004**$^{2}$ | −0.004$^{n}$ | −11.8$^{n}$ |
| GeoBPR | 0.034 | 0.148 | 0.754 | 0.263 | **0.636** | 0.074 | 2.158 | 0.697 | 0.513 | 0.114 | 0.752 | 992.6 |
| ΔGeoDiv (0.4) | −0.001$^{2}$ | −0.004$^{1}$ | **+0.011**$^{2}$ | **+0.009**$^{2}$ | −0.007$^{2}$ | −0.005$^{2}$ | **+0.845**$^{2}$ | **+0.138**$^{2}$ | **+0.158**$^{2}$ | −0.004$^{2}$ | **+0.038**$^{2}$ | **+188.6**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{2}$ | **+0.007**$^{2}$ | +0.005$^{2}$ | +0.005$^{2}$ | −0.001$^{n}$ | **+0.002**$^{2}$ | +0.633$^{2}$ | +0.111$^{2}$ | +0.091$^{2}$ | **+0.005**$^{2}$ | +0.010$^{2}$ | +116.2$^{2}$ |

$^{n,\ 1,\ 2}$The gain over (or loss under) the base recommender is statistically not significant (n), significant at $p < 0.05$ (1), or significant at $p < 0.01$ (2).
*Two-tailed paired t test* was used for the statistical significance test.
The value in () in Alg. column indicates the adopted diversification weight $\lambda$.

a slight accuracy decrease but does not achieve enough category diversification at the expense of accuracy loss.

**Effects of the base recommenders:** Although the trends of performance variations are similar in most cases, some of the variations are different owing to differences of the base recommenders. When we use GeoSoCa as the base recommender, the *EPC* and *AggDiv* increments are relatively small in comparison to the others. As we explained in Section 5.2.2, GeoSoCa recommends POIs concentrated in the most frequently visited areas of the target user. In addition, the content filtering of GeoSoCa filters out POIs that are popular but not in the preferred categories of each user. For instance, a famous ”Electronic shop” is popular by itself. But when we focus on each individual,

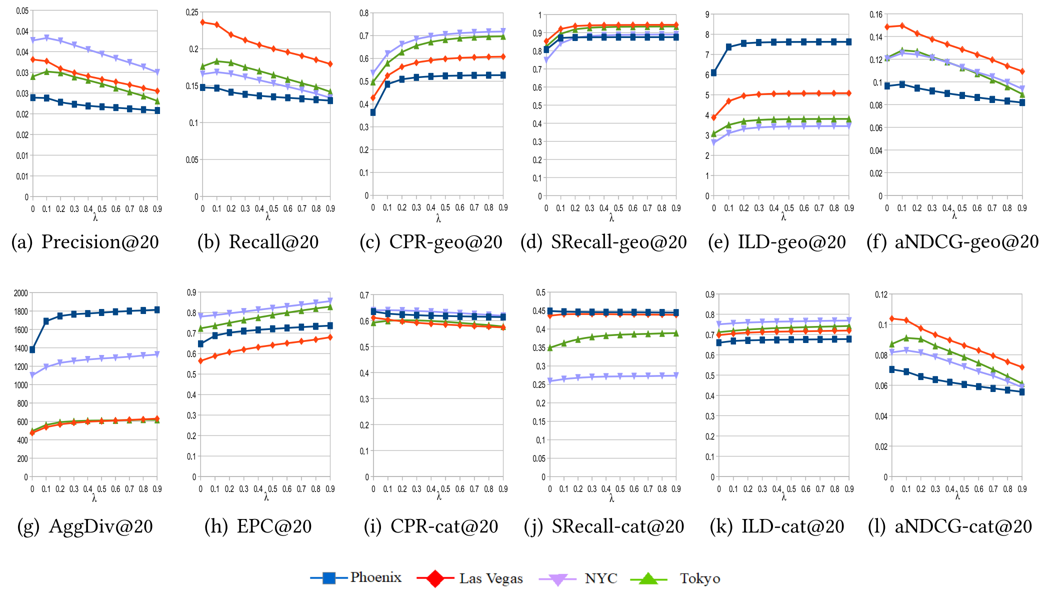
Table 13. Performance Improvement(+)/Decrease(−) with geoDiv and the Base Recommenders for Tokyo

| **Algorithm** | **$P$@5** | **$R$@5** | **$IL^c$@5** | **$SR^c$@5** | **$CP^c$@5** | **$aN^c$@5** | **$IL^g$@5** | **$SR^g$@5** | **$CP^g$@5** | **$aN^g$@5** | **$EP$@5** | **$AD$@5** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BPR | 0.048 | 0.078 | 0.671 | 0.165 | 0.549 | 0.060 | 2.795 | 0.607 | 0.350 | 0.082 | 0.672 | 226.2 |
| ΔGeoDiv (0.4) | −0.004$^{2}$ | −0.008$^{2}$ | **+0.030**$^{2}$ | **+0.007**$^{2}$ | −0.001$^{n}$ | −0.005$^{2}$ | **+0.874**$^{2}$ | **+0.108**$^{2}$ | **+0.141**$^{2}$ | −0.004$^{n}$ | **+0.083**$^{2}$ | **+62.8**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{2}$ | **+0.002**$^{1}$ | +0.002$^{n}$ | +0.004$^{2}$ | **+0.008**$^{2}$ | **+0.003**$^{2}$ | +0.303$^{2}$ | +0.047$^{2}$ | +0.055$^{2}$ | **+0.005**$^{2}$ | +0.019$^{2}$ | +25.4$^{2}$ |
| USG_Acc | 0.046 | 0.074 | 0.685 | 0.151 | 0.493 | 0.055 | 3.170 | 0.625 | 0.338 | 0.076 | 0.473 | 121.0 |
| ΔGeoDiv (0.4) | **+0.001**$^{n}$ | −0.000$^{n}$ | **+0.022**$^{2}$ | **+0.015**$^{2}$ | **+0.020**$^{2}$ | **+0.003**$^{1}$ | **+0.504**$^{2}$ | **+0.073**$^{2}$ | **+0.123**$^{2}$ | **+0.006**$^{2}$ | **+0.083**$^{2}$ | **+85.2**$^{2}$ |
| ΔGeoDivAcc (0.2) | **+0.001**$^{2}$ | **+0.001**$^{n}$ | +0.007$^{2}$ | +0.006$^{2}$ | +0.009$^{2}$ | **+0.003**$^{2}$ | +0.236$^{2}$ | +0.040$^{2}$ | +0.055$^{2}$ | +0.005$^{2}$ | +0.034$^{2}$ | +32.4$^{2}$ |
| GeoSoCa | 0.027 | 0.043 | 0.672 | 0.184 | 0.602 | 0.036 | 0.454 | 0.438 | 0.384 | 0.052 | 0.758 | 233.0 |
| ΔGeoDiv (0.2) | +0.000$^{n}$ | 0.000$^{n}$ | **+0.010**$^{2}$ | +0.001$^{n}$ | −0.002$^{n}$ | −0.002$^{n}$ | **+0.392**$^{2}$ | **+0.038**$^{2}$ | **+0.054**$^{2}$ | −0.002$^{n}$ | **+0.062**$^{2}$ | **+19.0**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{1}$ | **+0.003**$^{n}$ | +0.002$^{2}$ | **+0.002**$^{n}$ | **+0.004**$^{1}$ | **+0.001**$^{n}$ | +0.268$^{2}$ | +0.026$^{2}$ | +0.040$^{2}$ | **+0.002**$^{n}$ | +0.034$^{2}$ | +18.6$^{2}$ |
| GeoBPR | 0.042 | 0.066 | 0.698 | 0.163 | 0.512 | 0.052 | 2.363 | 0.592 | 0.378 | 0.077 | 0.670 | 211.2 |
| ΔGeoDiv (0.4) | −0.004$^{2}$ | −0.005$^{1}$ | **+0.023**$^{2}$ | **+0.007**$^{2}$ | **+0.009**$^{n}$ | −0.004$^{2}$ | **+1.067**$^{2}$ | **+0.113**$^{2}$ | **+0.120**$^{2}$ | −0.006$^{2}$ | **+0.076**$^{2}$ | **+49.0**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{n}$ | **+0.002**$^{n}$ | +0.003$^{1}$ | +0.003$^{2}$ | +0.004$^{1}$ | **+0.001**$^{n}$ | +0.442$^{2}$ | +0.051$^{2}$ | +0.044$^{2}$ | **+0.002**$^{n}$ | +0.016$^{2}$ | +19.4$^{1}$ |
| **Algorithm** | **$P$@10** | **$R$@10** | **$IL^c$@10** | **$SR^c$@10** | **$CP^c$@10** | **$aN^c$@10** | **$IL^g$@10** | **$SR^g$@10** | **$CP^g$@10** | **$aN^g$@10** | **$EP$@10** | **$AD$@10** |
| BPR | 0.038 | 0.119 | 0.692 | 0.245 | 0.577 | 0.072 | 2.937 | 0.725 | 0.420 | 0.101 | 0.697 | 336.0 |
| ΔGeoDiv (0.4) | −0.002$^{2}$ | −0.008$^{2}$ | **+0.025**$^{2}$ | **+0.020**$^{2}$ | +0.003$^{n}$ | −0.005$^{2}$ | **+0.843**$^{2}$ | **+0.128**$^{2}$ | **+0.170**$^{2}$ | −0.004$^{2}$ | **+0.068**$^{2}$ | **+96.0**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{2}$ | **+0.005**$^{2}$ | +0.004$^{2}$ | +0.006$^{2}$ | **+0.007**$^{2}$ | **+0.004**$^{2}$ | +0.382$^{2}$ | +0.064$^{2}$ | +0.069$^{2}$ | **+0.006**$^{2}$ | +0.016$^{2}$ | +42.4$^{2}$ |
| USG_Acc | 0.036 | 0.113 | 0.702 | 0.224 | 0.514 | 0.066 | 3.117 | 0.737 | 0.422 | 0.094 | 0.543 | 215.4 |
| ΔGeoDiv (0.4) | +0.001$^{1}$ | +0.001$^{n}$ | **+0.023**$^{2}$ | **+0.032**$^{2}$ | **+0.027**$^{2}$ | **+0.004**$^{2}$ | **+0.631**$^{2}$ | **+0.094**$^{2}$ | **+0.150**$^{2}$ | **+0.007**$^{2}$ | **+0.070**$^{2}$ | **+131.0**$^{2}$ |
| ΔGeoDivAcc (0.2) | **+0.002**$^{2}$ | **+0.005**$^{2}$ | +0.011$^{2}$ | +0.017$^{2}$ | +0.013$^{2}$ | **+0.004**$^{2}$ | +0.412$^{2}$ | +0.065$^{2}$ | +0.078$^{2}$ | **+0.007**$^{2}$ | +0.033$^{2}$ | +72.0$^{2}$ |
| GeoSoCa | 0.022 | 0.069 | 0.679 | 0.267 | 0.638 | 0.045 | 0.533 | 0.459 | 0.397 | 0.065 | 0.793 | 375.2 |
| ΔGeoDiv (0.2) | +0.001$^{2}$ | +0.002$^{1}$ | **+0.013**$^{2}$ | **+0.007**$^{2}$ | −0.004$^{1}$ | −0.001$^{n}$ | **+0.423**$^{2}$ | **+0.049**$^{2}$ | **+0.065**$^{2}$ | −0.000$^{n}$ | **+0.038**$^{2}$ | +26.8$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.002**$^{2}$ | **+0.005**$^{2}$ | +0.006$^{2}$ | +0.005$^{2}$ | **+0.003**$^{2}$ | **+0.002**$^{n}$ | +0.328$^{2}$ | +0.039$^{2}$ | +0.050$^{2}$ | **+0.003**$^{1}$ | +0.019$^{2}$ | **+28.6**$^{2}$ |
| GeoBPR | 0.033 | 0.104 | 0.714 | 0.250 | 0.545 | 0.063 | 2.421 | 0.684 | 0.439 | 0.094 | 0.701 | 328.0 |
| ΔGeoDiv (0.4) | −0.002$^{2}$ | −0.008$^{2}$ | **+0.019**$^{2}$ | **+0.017**$^{2}$ | **+0.010**$^{1}$ | −0.004$^{2}$ | **+1.096**$^{2}$ | **+0.142**$^{2}$ | **+0.151**$^{2}$ | −0.007$^{2}$ | **+0.060**$^{2}$ | **+74.6**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.004**$^{2}$ | +0.004$^{2}$ | +0.008$^{2}$ | +0.007$^{2}$ | **+0.002**$^{2}$ | +0.633$^{2}$ | +0.084$^{2}$ | +0.065$^{2}$ | **+0.003**$^{2}$ | +0.014$^{2}$ | +39.8$^{2}$ |
| **Algorithm** | **$P$@20** | **$R$@20** | **$IL^c$@20** | **$SR^c$@20** | **$CP^c$@20** | **$aN^c$@20** | **$IL^g$@20** | **$SR^g$@20** | **$CP^g$@20** | **$aN^g$@20** | **$EP$@20** | **$AD$@20** |
| BPR | 0.029 | 0.176 | 0.712 | 0.350 | 0.592 | 0.087 | 3.076 | 0.824 | 0.497 | 0.121 | 0.723 | 495.8 |
| ΔGeoDiv (0.4) | −0.001$^{2}$ | −0.006$^{2}$ | **+0.020**$^{2}$ | **+0.033**$^{2}$ | +0.006$^{1}$ | −0.005$^{2}$ | **+0.706**$^{2}$ | **+0.107**$^{2}$ | **+0.176**$^{2}$ | −0.004$^{2}$ | **+0.052**$^{2}$ | **+113.2**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.007**$^{2}$ | +0.007$^{2}$ | +0.012$^{2}$ | **+0.007**$^{2}$ | **+0.004**$^{2}$ | +0.431$^{2}$ | +0.070$^{2}$ | +0.083$^{2}$ | **+0.006**$^{2}$ | +0.013$^{2}$ | +66.0$^{2}$ |
| USG_Acc | 0.028 | 0.172 | 0.721 | 0.325 | 0.529 | 0.081 | 3.129 | 0.824 | 0.513 | 0.116 | 0.610 | 397.6 |
| ΔGeoDiv (0.4) | +0.001$^{2}$ | +0.005$^{1}$ | **+0.017**$^{2}$ | **+0.043**$^{2}$ | **+0.028**$^{2}$ | +0.004$^{2}$ | **+0.566**$^{2}$ | **+0.077**$^{2}$ | **+0.151**$^{2}$ | **+0.008**$^{2}$ | **+0.050**$^{2}$ | **+142.0**$^{2}$ |
| ΔGeoDivAcc (0.2) | **+0.002**$^{2}$ | **+0.008**$^{2}$ | +0.011$^{2}$ | +0.030$^{2}$ | +0.016$^{2}$ | **+0.005**$^{2}$ | +0.488$^{2}$ | +0.067$^{2}$ | +0.096$^{2}$ | **+0.008**$^{2}$ | +0.026$^{2}$ | +104.4$^{2}$ |
| GeoSoCa | 0.017 | 0.106 | 0.693 | 0.368 | 0.660 | 0.055 | 0.614 | 0.480 | 0.411 | 0.081 | 0.820 | 582.8 |
| ΔGeoDiv (0.2) | **+0.001**$^{2}$ | +0.005$^{1}$ | **+0.007**$^{2}$ | **+0.009**$^{2}$ | −0.002$^{1}$ | +0.001$^{n}$ | **+0.325**$^{2}$ | **+0.045**$^{2}$ | **+0.064**$^{2}$ | +0.001$^{n}$ | **+0.024**$^{2}$ | +18.0$^{1}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.006**$^{2}$ | +0.003$^{2}$ | +0.005$^{2}$ | **+0.004**$^{2}$ | **+0.003**$^{2}$ | +0.262$^{2}$ | +0.037$^{2}$ | +0.051$^{2}$ | **+0.003**$^{2}$ | +0.011$^{n}$ | **+25.6**$^{2}$ |
| GeoBPR | 0.026 | 0.154 | 0.732 | 0.370 | 0.569 | 0.077 | 2.471 | 0.765 | 0.505 | 0.114 | 0.730 | 509.8 |
| ΔGeoDiv (0.4) | 0.000$^{n}$ | −0.003$^{n}$ | **+0.011**$^{2}$ | **+0.020**$^{2}$ | **+0.007**$^{1}$ | −0.003$^{2}$ | **+0.892**$^{2}$ | **+0.112**$^{2}$ | **+0.157**$^{2}$ | −0.006$^{2}$ | **+0.043**$^{2}$ | **+81.0**$^{2}$ |
| ΔGeoDivAcc (0.1) | **+0.001**$^{2}$ | **+0.007**$^{2}$ | +0.004$^{2}$ | +0.012$^{2}$ | **+0.007**$^{2}$ | **+0.003**$^{2}$ | +0.719$^{2}$ | +0.096$^{2}$ | +0.088$^{2}$ | **+0.003**$^{2}$ | +0.011$^{2}$ | +58.0$^{2}$ |

$^{n,\ 1,\ 2}$The gain over (or loss under) the base recommender is statistically not significant (n), significant at $p < 0.05$ (1), or significant at $p < 0.01$ (2).
*Two-tailed paired t test* was used for the statistical significance test.
The value in () in Alg. column indicates the adopted diversification weight $\lambda$.

most users do not frequently visit electronic shops compared to restaurants. Electronic shops are likely to be filtered out by the content filtering with high probability. As a consequence, GeoSoCa achieves high *EPC*. Applying the proposed method causes GeoSoCa to recommend more "popular" POIs that are located in other areas. As a consequence, the *EPC* and *AggDiv* increments are limited.

With USG_Acc, GeoBPR, and BPR base recommenders that do not generate extremely geographically skewed recommendations, the proposed method achieves substantial improvement in geo-diversity, novelty, and coverage with tolerable loss of accuracy. Because the potential diversity improvement of reordering algorithms depends on the base recommendation list, moderately geographically biased algorithms such as USG and GeoBPR may show acceptable performance




(a) Precision@20 (b) Recall@20 (c) CPR-geo@20 (d) SRecall-geo@20 (e) ILD-geo@20 (f) aNDCG-geo@20

(g) AggDiv@20 (h) EPC@20 (i) CPR-cat@20 (j) SRecall-cat@20 (k) ILD-cat@20 (l) aNDCG-cat@20

(m) Legend

Fig. 5. Effects of the diversification weight $\lambda$ (Base recommender: BPR).

when coupled with the proposed method. Similarly, one reason for BPR's superior performance in evaluation is that, in addition to its superiority in accuracy, BPR can generate a moderately geographically biased base recommendation list, because we limit the recommendation candidate POIs to the POIs located in the user's frequently visited areas. From the results, it is desirable to select an algorithm that does not generate extremely skewed recommendations as a base recommender.

**Performance in category diversification:** Our proposed method focuses on diversification of POI locations, so we cannot detect consistent improvement in POI category diversity. $ILD^{cat}$ slightly increases for all evaluated cases, because the proposed method introduces some POIs in rare categories not frequently placed in the top part of the recommendation list by the base recommenders. $CPR^{cat}$ and $SRecall^{cat}$ do not show a consistent improvement or a decrement ternd over the four cities. Unlike $ILD^{cat}$ that measures distance between categories, $CPR^{cat}$ and $SRecall^{cat}$ consider each individual's preference to categories. For example, to achieve a high value in $CPR^{cat}$ or $SRecall^{cat}$, a recommender should recommend POIs from categories previously visited by the target user according to the requirements specified by each metric. The proposed method does not explicitly consider user-based category diversity.

**Effect of the diversification weight:** Figure 5 shows the performance of our proposed method with respect to the value of diversification weight $\lambda$ when BPR is selected as the base recommender. We plot the performance of the BPR-base-recommender-configuration, because the configuration achieves the best overall performance.

*Precision*@20 and *Recall*@20, at $\lambda = 0.1$, show a small increment in NYC and Tokyo and slight decrement in Phoenix and LasVegas. Then the values in the metrics linearly decrease as the value of $\lambda$ increases. $CPR^{geo}$@20, $SRecall^{geo}$@20, $ILD^{geo}$@20, and *AggDiv*@20 are greatly improved in the range of $\lambda = [0.0, 0.2]$ and the improvement begins to saturate above $\lambda = 0.2$. *EPC*@20 linearly increases as $\lambda$ increases. The results re-confirm that small values of $\lambda$ such as 0.1 can improve geo-diversity, coverage, and novelty with negligible accuracy loss. Thus, $\lambda = [0.1, 0.2]$ indicates a proper range to achieve a balance between accuracy metrics and the other metrics.

Increment or decrement trends w.r.t. $\lambda$ increment are not consistent among the category diversification metrics. One of personalized category diversification metrics, $CPR^{cat}$@20, decreases as $\lambda$ increases. The other metric, $SRecall^{cat}$@20, show different trends based on LBSN services. $SRecall^{cat}$@20 decrease as $\lambda$ increases for the Yelp dataset while the values slightly increase as $\lambda$ increments with Foursquare dataset. As discussed in "Performance in category diversification" in this section, $ILD^{cat}$@20 increases as $\lambda$ increases but the amount is small. The observed inconsistency is caused by the lack of personal category diversity awareness of our proposed methods.

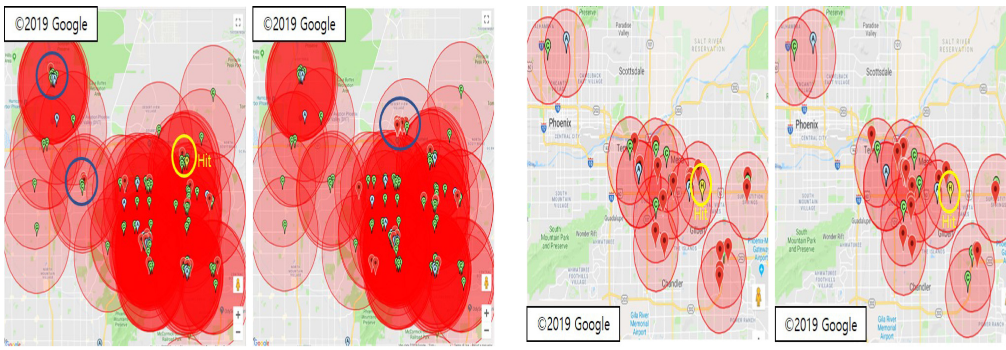
$\alpha$-$NDCG^{geo}$ shows a small increment at $\lambda = 0.1$, then the value in the metrics linearly decrease as the value of $\lambda$ increases in all four cities. $\alpha$-$NDCG^{cat}$ shows two different patterns by dataset. NYC and Tokyo from Foursquare dataset, similarly to $\alpha$-$NDCG^{geo}$, $\alpha$-$NDCG^{cat}$ shows a small increment at $\lambda = 0.1$, then the value in the metric linearly decreases as the value of $\lambda$ increases while constant decrement is observed in Phoenix and Las Vegas from Yelp dataset. Since we can roughly consider $\alpha$-$NDCG$ in this article as a kind of combination of $Recall$ and $SRecall$, the observed trends follow the combined trend of the two metrics.

*5.3.3 Case Study.* In this subsection, we discuss three cases to obtain a rough picture of user perception regarding the proposed method and to discuss the benefits and drawbacks of our proposed method. Figure 6 shows the results of 20 POI recommendations to User1, User2, and User3 in Phoenix. We plot the results for the proposed method with the BPR base recommender and the results from BPR (Base recommender only) side by side to compare the differences.

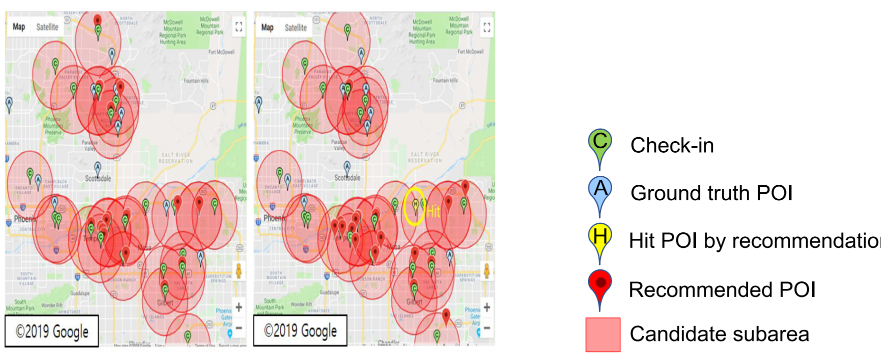
As shown in Figure 6(a) and (b), User1 has two densely visited area groups. In addition to the largest group, the proposed method recommends POIs from the other group and from an middle area between the two groups (the blue circles in Figure 6(a)) while BPR recommends only POIs from the largest group. Especially, many POIs located on the edge of the large densely visited area group (the blue circle in Figure 6(b)) are recommended by BPR. The recommended POIs are dispersed by the proposed method. This is exactly what we expected from the proposed method, and User1 receives a POI recommendation that covers more of his/her frequently visited areas. Figure 6(c) and (d) show a case that is almost a tie. Although the POIs recommended by the proposed method seem to provide slightly better geo-diversity, the POIs recommended by BPR also seem to achieve good geo-diversity. One possible explanation for this is that the user's visit pattern is similar to the publics' visit patterns in those areas.

An unsuccessful case for the proposed method is shown in Figure 6(e) and (f). The recommendation generated by BPR can retrieve a ground-truth POI that cannot be retrieved by the proposed method. The retrieved ground-truth POI is located a relatively sparsely visited area (The yellow circle in Figure 6(f)). The proposed method suffers difficulty to retrieve the ground-truth POI, because the density of visit is relatively sparse in the area. One drawback of our proposed method is that we use observed check-ins of the target user as a diversification reference, and under the current configuration, the method rarely cares about POIs located in sparsely visited areas.

*5.3.4 Performance of Category Diversification Algorithms.* In this subsection, we discuss the difficulty of geo-diversity improvement by adopting existing category diversification methods. Existing diversification methods mainly focus on category diversification of the recommended POIs. PM2 [16] is a state-of-the-art diversification method that diversifies the recommendations to achieve high $CPR^{cat}$, while ILDiv is a popular diversification method that diversifies the recommendation to achieve high $ILD^{cat}$ (the objective function of ILDiv is found in Equation (2) of Reference [51], and we adopt Equation (22) in this article as $dist(i, j)$). Because category diversification methods do not directly consider the locations of POIs, it is difficult for them to greatly




(a) Proposed for User1 (Successful) (b) BPR for User1 (Unsuccessful) (c) Proposed for User2 (Tie) (d) BPR for User2 (Tie)




(e) Proposed for User3 (Unsuccessful) (f) BPR for User3 (Successful) (g) Legend

Fig. 6. Case study in Phoenix: The proposed method (BPR+GeoDiv(0.3)) V.S. BPR.

Table 14. Parameters for Category Diversification Algorithms

| Algorithm | Parameters |
|---|---|
| PM2 ($x$) [16] | Weight of main quotient $\lambda = x$. |
| ILDiv ($x$) (Equation (2) of [51]) | Diversification weight $\lambda = x$. |

improve geo-diversity. This is also supported by our performance evaluation of PM2 and ILDiv. The settings and the evaluation results are shown in Table 14 and Table 15 respectively. The reported performance is achieved by applying each diversification method to a base recommendation list of size $n = 4k (k = 20)$ generated by BPR.

**PM2:** With some loss in *Prediction* and *Recall*, PM2 (0.8) achieves the best $SRecall^{cat}$, $CPR^{cat}$, *EPC*, and *AggDiv*. PM2 (1.0) shows no loss in *Prediction*@20 and *Recall*@20 in exchange for less improvement in the other four metrics. The results show that properly tuned PM2 achieves user preference aware category diversification, novelty, and coverage with some prediction accuracy loss. PM2 (0.8) achieves decreased $ILD^{cat}$ compared to BPR, and PM2 (1.0) shows small deviations in $ILD^{cat}$ values from those of BPR. The $ILD^{cat}$ decrement of PM2 indicates that user preference for categories exhibits trends that are not totally random. Because user preferences are biased to some categories, reordered POIs follow the user preference bias, therefore reducing $ILD^{cat}$, which reflects a type of randomness of category appearance in the recommended list.

Table 15. Performance Improvement(+)/Decrease(−) of Category Diversification Algorithms from the Base Recommenders

| Ct[3] | Algorithm | $P$@20 | $R$@20 | $IL^c$@20 | $SR^c$@20 | $CP^c$@20 | $aN^c$@20 | $IL^g$@20 | $SR^g$@20 | $CP^g$@20 | $aN^g$@20 | $EP$@20 | $AD$@20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | BPR | **0.024** | 0.148 | 0.660 | 0.449 | 0.635 | 0.070 | 6.094 | 0.807 | 0.363 | **0.096** | 0.648 | 1380.2 |
| | ΔPM2 (1.0) | **+0.000**[2] | **+0.001**[1] | +0.002[2] | +0.045[2] | +0.095[2] | +0.005[2] | +0.109[2] | +0.005[2] | +0.003[2] | −0.003[2] | +0.018[2] | +71.2[2] |
| | ΔPM2 (0.8) | −0.003[2] | −0.022[2] | −0.119[2] | **+0.178**[2] | **+0.256**[2] | **+0.017**[2] | **+1.123**[2] | **+0.044**[2] | **+0.016**[2] | −0.017[2] | **+0.043**[2] | **+215.6**[2] |
| | ΔILDiv (0.5) | −0.004[2] | −0.027[2] | +0.122[2] | +0.083[2] | −0.074[2] | −0.018[2] | +0.503[2] | +0.027[2] | +0.008[2] | −0.013[2] | +0.017[2] | −21.0[1] |
| | ΔILDiv (0.9) | −0.004[2] | −0.029[2] | **+0.123**[2] | +0.084[2] | −0.076[2] | −0.018[2] | +0.531[2] | +0.028[2] | +0.008[2] | −0.013[2] | +0.018[2] | −30.4[1] |
| L | BPR | **0.033** | 0.236 | 0.697 | 0.436 | 0.611 | 0.104 | 3.871 | 0.853 | 0.427 | **0.149** | 0.565 | 475.4 |
| | ΔPM2 (1.0) | **+0.000**[1] | **+0.001**[n] | +0.001[2] | +0.059[2] | +0.108[2] | +0.009[2] | +0.076[2] | +0.006[2] | +0.009[2] | −0.006[2] | +0.031[2] | +41.2[2] |
| | ΔPM2 (0.8) | −0.005[2] | −0.042[2] | −0.088[2] | **+0.217**[2] | **+0.269**[2] | **+0.027**[2] | **+0.698**[2] | **+0.040**[2] | **+0.025**[2] | −0.028[2] | **+0.090**[2] | **+120.2**[2] |
| | ΔILDiv (0.5) | −0.007[2] | −0.046[2] | +0.098[2] | +0.089[2] | −0.076[2] | −0.026[2] | +0.467[2] | +0.030[2] | +0.011[2] | −0.019[2] | +0.003[n] | −8.0[1] |
| | ΔILDiv (0.9) | −0.007[2] | −0.055[2] | **+0.102**[2] | +0.092[2] | −0.079[2] | −0.029[2] | +0.501[2] | +0.032[2] | +0.012[2] | −0.022[2] | +0.004[n] | −12.6[2] |
| N | BPR | **0.038** | 0.166 | 0.750 | 0.259 | 0.640 | 0.082 | 2.623 | 0.748 | 0.535 | **0.121** | 0.781 | 1100.2 |
| | ΔPM2 (1.0) | **+0.000**[n] | **+0.001**[n] | +0.002[2] | +0.039[2] | +0.121[2] | +0.004[2] | +0.033[2] | +0.003[2] | +0.005[2] | −0.008[2] | +0.015[2] | +40.2[2] |
| | ΔPM2 (0.8) | −0.005[2] | −0.023[2] | −0.036[2] | **+0.208**[2] | **+0.284**[2] | **+0.024**[2] | **+0.178**[2] | **+0.021**[2] | **+0.013**[2] | −0.023[2] | **+0.053**[2] | **+129.0**[2] |
| | ΔILDiv (0.5) | −0.004[2] | −0.017[2] | +0.049[2] | +0.084[2] | −0.038[2] | −0.012[2] | +0.083[2] | +0.010[2] | +0.004[2] | −0.009[2] | −0.001[n] | −41.6[2] |
| | ΔILDiv (0.9) | −0.005[2] | −0.020[2] | **+0.050**[2] | +0.087[2] | −0.040[2] | −0.013[2] | +0.095[2] | +0.011[2] | +0.004[2] | −0.010[2] | −0.001[n] | −52.8[2] |
| T | BPR | **0.029** | 0.176 | 0.712 | 0.350 | 0.592 | 0.087 | 3.076 | 0.824 | 0.497 | **0.121** | 0.723 | 495.8 |
| | ΔPM2 (1.0) | **+0.000**[n] | **+0.001**[n] | −0.000[n] | +0.089[2] | +0.171[2] | +0.005[2] | +0.026[2] | +0.004[2] | +0.008[2] | −0.008[2] | +0.035[2] | +31.4[2] |
| | ΔPM2 (0.8) | −0.003[2] | −0.016[2] | −0.087[2] | **+0.259**[2] | **+0.329**[2] | **+0.024**[2] | **+0.150**[2] | +0.013[2] | **+0.015**[2] | −0.020[2] | **+0.082**[2] | **+105.4**[2] |
| | ΔILDiv (0.5) | −0.006[2] | −0.031[2] | +0.083[2] | +0.141[2] | −0.083[2] | −0.020[2] | +0.063[2] | +0.014[2] | +0.005[2] | −0.016[2] | −0.007[2] | −52.0[2] |
| | ΔILDiv (0.9) | −0.008[2] | −0.045[2] | **+0.088**[2] | +0.150[2] | −0.094[2] | −0.025[2] | +0.081[2] | **+0.015**[2] | +0.005[2] | −0.021[2] | −0.008[2] | −71.6[2] |

[n, 1, 2]The gain over (or loss under) the base recommender is statistically not significant (n), significant at $p < 0.05$ (1), or significant at $p < 0.01$ (2).
*Two-tailed paired t test* was used for the statistical significance test.
The value in () in Algorithm column indicates the adopted parameter specific to each algorithm.
[3]City. P : Phoenix, L : Las Vegas, N: New York, T: Tokyo.

Although PM2 increases $SRecall^{geo}$ and $CPR^{geo}$, the improvement is limited. $ILD^{geo}$ improvement depends on the dataset. $ILD^{geo}$ improvements by PM2 (0.8) for Phoenix and Las Vegas are smaller but the size of improvements is more than 60% of the $ILD^{geo}$ improvements achieved by the proposed method, while the improvements by PM2 (0.8) are less than 25% of the $ILD^{geo}$ improvements achieved by the proposed method for New York and Tokyo. One possible explanation is the difference of geographic distributions of POIs in each dataset. New York and Tokyo are bigger cities and have higher populations than Phoenix and Las Vegas. In addition, check-in data in the Foursquare dataset are generated by simple clicks in an application while check-in data in the Yelp dataset are generated by writing reviews that involve more user time. As a result, the density of POIs in a given geographic area is higher in New York and Tokyo than in Phoenix and Las Vegas. Because of the POI density difference, POIs that satisfy PM2's diversification goal are found over a wider area in Phoenix and Las Vegas, while sufficient numbers of POIs are found in narrower areas in New York and Tokyo. The results shown for $ILD^{geo}$, $SRecall^{geo}$, and $CPR^{geo}$ imply that PM2 can introduce randomness of POI locations to the recommended POI list but the introduction depends on the dataset. It is difficult for PM2 to improve geo-diversity, because PM2 does not explicitly consider an individual's movement patterns in the process of diversification.

**ILDiv:** As expected, ILDiv shows significant improvement in $ILD^{cat}$ with some loss in *Precision* and *Recall*. ILDiv also achieves improvement in $SRecall^{cat}$ while $CPR^{cat}$ is decreased, because some randomness in recommended POI categories introduced to ILDiv does not impede finding some categories related to POIs visited by the target user, but it cannot improve the category proportionality required by $CPR^{cat}$. Similar patterns of $ILD^{geo}$, $SRecall^{geo}$, and $CPR^{geo}$ to those of PM2 indicate that ILDiv also has difficulty achieving geo-diversity. For novelty and coverage, compared to PM2, the improvement in *EPC* is limited and the performance of *AggDiv* is deteriorated. This ineffectiveness indicates that intra-list diversification of POI categories can be achieved without careful consideration regarding novelty and coverage. In other words, high $ILD^{cat}$ can be achieved by recommending the relatively popular POIs of different categories that are located in a small number of concentrated areas.

**Diversity confusion as future work:** The results we have seen in this subsection indicate that it is difficult to achieve category and geo-diversification simultaneously by either the proposed method or the other category diversification methods. A simple method of achieving both types of diversity is by using weighted sums or multiplication of the diversification objective values. Although we may achieve acceptable diversity of POI category and location with tolerable accuracy loss with proper tuning, we still do not have a clear picture on how to balance the diversity of categories and locations. The balanced diversity achieved by the combination of existing diversification methods may take on different shapes when we analyze the recommended POIs in detail. For instance, even if we achieved high $CPR^{cat}$ and $CPR^{geo}$ by a combination of PM2 and the proposed method, the category distribution of recommended POIs over geographic areas can have variations. Consider the following two cases for a user who has two separate frequently visited areas. In the first case, the recommended POIs located in each area are included in the small number of categories, and the recommendation achieves diverse categories in total. In the second case, the recommended POIs located in any area are included in similarly diverse categories. We cannot simply say that either of the cases is better than the other, because the answer depends on each user's preference. Implementation also has challenges to be answered. For example, inference of user category preferences in a relatively small subarea in sparse data environments is one challenge. We consider this type of problem as future work.

*5.3.5 Application and Limitation.* In this section, we discuss an application of the proposed method and then describe a limitation of this article. One possible application is a personalized POI promotion. Since the proposed method is designed to recommend POIs from "diverse" locations in the user's "frequently visited" areas, from users' point of view, preferable POIs of easy access are informed. From POI owners' point of view, his/her POI can be revealed to the users of higher visit probability. In addition, even if the POIs are located in relatively less popular areas in the city, they have more chances to be revealed to the users.

This kind of application often requires a quick serving for a massive number of users via mobile hand-held devices such as smartphones. If a target service allows pre-calculation, then the proposed method is relatively easily implemented by distributed computing clusters. The users are able to be distributed over computing nodes in the cluster to pre-calculate the recommendations simultaneously, because the re-ordering process does not require information of the other users. Unfortunately, many applications need more real-time interactions with users. For instance, users may want a manual control to the weight of each frequently visited areas or a smart service may need real-time contexts such as recommendation time and current user location for each request. Adopting the proposed method to real-time environment introduces some challenge. In addition to the computing cost of the base recommender, $O(nk)$ of the computing cost is required for re-ordering process, where $n$ and $k$ indicate the number of recommendation candidates and the number of recommended POIs, respectively. In real service, additional $O(nk)$ computing cost is still burdensome. If the number of target users require concurrent processing becomes bigger, then the cost (number) of serving machines easily becomes heavier for most of small companies such as start-ups. Developing a more lightweight diversification method with a similar diversification performance to re-ordering methods is one challenge.

One limitation of this article is that the effect of proposed method is evaluated only with the data of big cities. It will be more valuable if we can evaluate the performance of our proposed method in smaller, mid-sized cities, which are harder to predict. However, it was difficult for us to collect enough evaluation data for small cities. We collected the Foursquare data via Twitter API, but with less than 1% of Twitter data provided by the API, we could not collected enough evaluation data for small cities. Although Yelp dataset contains data from small cities, their sizes are also far smaller compared to those of the two big cities, Phoenix and Las Vegas. We leave the evaluation on smaller sized cities as a future work.

## 6 CONCLUSION

We introduced a novel geographical diversity (geo-diversity) concept as another evaluation perspective for POI recommendation and proposed a metric to measure geo-diversity. The evaluation using real LBSN datasets showed that existing POI recommender algorithms are not capable of greatly improving geo-diversity. We proposed a reordering-based diversification method that improves geo-diversity when added to existing POI recommenders. The proposed method is easily applicable to existing POI recommender systems. The evaluation showed that the proposed method greatly improved geo-diversity with tolerable loss of accuracy and even improved the geo-diversity when we tuned models for negligible accuracy loss. More specifically, the proposed method consistently increased geo-diversity ($CPR^{geo}$@20) by 0.05 to 0.12 using four state-of-the-art POI recommenders with negligible prediction accuracy loss (0.005 loss to 0.007 increment in $Recall$@20) and by 0.06 to 0.18 with tolerable prediction accuracy loss (0.024 loss to 0.005 increment in $Recall$@20). In addition, the proposed method also improves the novelty and coverage of recommended POIs.

Because our proposed method does not explicitly consider POI category diversity, it cannot consistently improve this metric. Although POI category diversity is improved by existing diversification methods, the methods have difficulty improving geo-diversity. One of our future works will be a study regarding appropriate integration of category diversity and geo-diversity for POI recommendation.

## REFERENCES


[1] Panagiotis Adamopoulos and Alexander Tuzhilin. 2014. On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14)*. ACM, New York, NY, 153–160. DOI: https://doi.org/10.1145/2645710.2645752

[2] Mohammad Aliannejadi and Fabio Crestani. 2017. Venue appropriateness prediction for personalized context-aware venue suggestion. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17)*. ACM, New York, NY, 1177–1180. DOI: https://doi.org/10.1145/3077136.3080754

[3] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL'12)*. ACM, New York, NY, 199–208. DOI: https://doi.org/10.1145/2424321.2424348

[4] Ramesh Baral and Tao Li. 2018. Exploiting the roles of aspects in personalized POI recommender systems. *Data Min. Knowl. Discov.* 32, 2 (Mar. 2018), 320–343. DOI: https://doi.org/10.1007/s10618-017-0537-7

[5] Ramesh Baral, XiaoLong Zhu, S. S. Iyengar, and Tao Li. 2018. ReEL: Review aware explanation of location recommendation. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP'18)*. ACM, New York, NY, 23–32. DOI: https://doi.org/10.1145/3209219.3209237

[6] Fabiano Belém, Rodrygo Santos, Jussara Almeida, and Marcos Gonçalves. 2013. Topic diversity in tag recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 141–148. DOI: https://doi.org/10.1145/2507157.2507184

[7] Idir Benouaret and Dominique Lenne. 2016. A package recommendation framework for trip planning activities. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys'16)*. ACM, New York, NY, 203–206. DOI: https://doi.org/10.1145/2959100.2959183

[8] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98).* ACM, New York, NY, 335–336. DOI: https://doi.org/10.1145/290941.291025

[9] Luis Castillo, Eva Armengol, Eva Onaindía, Laura Sebastiá, Jesús González-Boticario, Antonio Rodríguez, Susana Fernández, Juan D. Arias, and Daniel Borrajo. 2008. Samap: An user-oriented adaptive system for planning tourist visits. *Exp. Syst. Appl.* 34, 2 (Feb. 2008), 1318–1332. DOI: https://doi.org/10.1016/j.eswa.2006.12.029

[10] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-aware hierarchical point-of-interest embedding model for successive POI recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18).* AAAI Press, 3301–3307. http://dl.acm.org/citation.cfm?id=3304222.3304226

[11] Xuefeng Chen, Yifeng Zeng, Gao Cong, Shengchao Qin, Yanping Xiang, and Yuanshun Dai. 2015. On information coverage for location category based point-of-interest recommendation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15).* AAAI Press, 37–43. http://dl.acm.org/citation.cfm?id=2887007.2887013

[12] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12).* AAAI Press, 17–23. http://dl.acm.org/citation.cfm?id=2900728.2900731

[13] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13).* AAAI Press, 2605–2611. http://dl.acm.org/citation.cfm?id=2540128.2540504

[14] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11).* ACM, New York, NY, 1082–1090. DOI: https://doi.org/10.1145/2020408.2020579

[15] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08).* ACM, New York, NY, 659–666. DOI: https://doi.org/10.1145/1390334.1390446

[16] Van Dang and W. Bruce Croft. 2012. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12).* ACM, New York, NY, 65–74. DOI: https://doi.org/10.1145/2348283.2348296

[17] Romain Deveaud, M-Dyaa Albakour, Craig Macdonald, and Iadh Ounis. 2015. Experiments with a venue-centric model for personalisedand time-aware venue suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM'15).* ACM, New York, NY, 53–62. DOI: https://doi.org/10.1145/2806416.2806484

[18] Rosie Dunford, Quanrong Su, and Ekraj Tamang. 2014. The Pareto principle. *Plymouth Stud. Sci.* 7, 1 (2014), 140–148. https://bcur.org/journals/index.php/TPSS/article/view/408.

[19] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical latent representation for predicting future visitors. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*102–108.

[20] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15).* AAAI Press, 2069–2075. http://dl.acm.org/citation.cfm?id=2832415.2832536

[21] Daniel Fleder and Kartik Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Manage. Sci.* 55, 5 (May 2009), 697–712. DOI: https://doi.org/10.1287/mnsc.1080.0974

[22] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13).* ACM, New York, NY, 93–100. DOI: https://doi.org/10.1145/2507157.2507182

[23] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-aware point of interest recommendation on location-based social networks. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15).* AAAI Press, 1721–1727. http://dl.acm.org/citation.cfm?id=2886521.2886559

[24] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10).* ACM, New York, NY, 257–260. DOI: https://doi.org/10.1145/1864708.1864761

[25] Jean-Benoit Griesner, Talel Abdessalem, and Hubert Naacke. 2015. POI recommendation: Towards fused matrix factorization with geographical and temporal influences. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys'15).* ACM, New York, NY, 301–304. DOI: https://doi.org/10.1145/2792838.2799679

[26] Qing Guo, Zhu Sun, Jie Zhang, Qi Chen, and Yin-Leng Theng. 2017. Aspect-aware point-of-interest recommendation with geo-social influence. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP'17).* ACM, New York, NY, 17–22. DOI: https://doi.org/10.1145/3099023.3099066

[27] Jungkyu Han and Hayato Yamana. 2017. Geographical diversification in POI recommendation: Toward improved coverage on interested areas. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys'17)*. ACM, New York, NY, 224–228. DOI: https://doi.org/10.1145/3109859.3109884

[28] Mengyue Hang, Ian Pytlarz, and Jennifer Neville. 2018. Exploring student check-in behavior for improved point-of-interest prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'18)*. ACM, New York, NY, 321–330. DOI: https://doi.org/10.1145/3219819.3219902

[29] Jing He, Xin Li, Lejian Liao, Dandan Song, and William K. Cheung. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 137–143. http://dl.acm.org/citation.cfm?id=3015812.3015833

[30] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53. DOI: https://doi.org/10.1145/963770.963772

[31] Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-N recommendation—Analysis and evaluation. *ACM Trans. Internet Technol.* 10, 4, Article 14 (Mar. 2011), 30 pages. DOI: https://doi.org/10.1145/1944339.1944341

[32] Neil J. Hurley. 2013. Personalised ranking with diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 379–382. DOI: https://doi.org/10.1145/2507157.2507226

[33] Onur Küçüktunç, Erik Saule, Kamer Kaya, and Ümit V. Çatalyürek. 2014. Diversifying citation recommendations. *ACM Trans. Intell. Syst. Technol.* 5, 4, Article 55 (Dec. 2014), 21 pages. DOI: https://doi.org/10.1145/2668106

[34] Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshide, Noriko Takaya, and Ko Fujimura. 2013. Geo topic model: Joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM'13)*. ACM, New York, NY, 375–384. DOI: https://doi.org/10.1145/2433396.2433444

[35] Huayu Li, Yong Ge, Richang Hong, and Hengshu Zhu. 2016. Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, New York, NY, 975–984. DOI: https://doi.org/10.1145/2939672.2939767

[36] Huayu Li, Richang Hong, Shiai Zhu, and Yong Ge. 2015. Point-of-interest recommender systems: A separate-space perspective. In *Proceedings of the 2015 IEEE International Conference on Data Mining*. 231–240. DOI: https://doi.org/10.1109/ICDM.2015.27

[37] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: A scalable two-stage personalized news recommendation system. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, NY, 125–134. DOI: https://doi.org/10.1145/2009916.2009937

[38] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. ACM, New York, NY, 433–442. DOI: https://doi.org/10.1145/2766462.2767722

[39] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*. ACM, New York, NY, 831–840. DOI: https://doi.org/10.1145/2623330.2623638

[40] Yanchi Liu, Chuanren Liu, Bin Liu, Meng Qu, and Hui Xiong. 2016. Unified point-of-interest recommendation with temporal interval assessment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, New York, NY, 1015–1024. DOI: https://doi.org/10.1145/2939672.2939773

[41] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proc. VLDB Endow.* 10, 10 (Jun. 2017), 1010–1021. DOI: https://doi.org/10.14778/3115404.3115407

[42] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM'14)*. ACM, New York, NY, 739–748. DOI: https://doi.org/10.1145/2661829.2662002

[43] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A personalised ranking framework with multiple sampling criteria for venue recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM'17)*. ACM, New York, NY, 1469–1478. DOI: https://doi.org/10.1145/3132847.3132985

[44] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems (CHI EA'06)*. ACM, New York, NY, 1097–1101. DOI: https://doi.org/10.1145/1125451.1125659

[45] Katja Niemann and Martin Wolpers. 2013. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. ACM, New York, NY, 955–963. DOI: https://doi.org/10.1145/2487575.2487656

[46] Michael J. Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.* 13, 5 (01 Dec. 1999), 393–408. DOI: https://doi.org/10.1023/A:1006544522159

[47] Pearl Pu, Li Chen, and Rong Hu. 2012. Evaluating recommender systems from the user's perspective: Survey of the state of the art. *User Model. User-Adapt. Interact.* 22, 4–5 (Oct. 2012), 317–355. DOI: https://doi.org/10.1007/s11257-011-9115-7

[48] Tieyun Qian, Bei Liu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2019. Spatiotemporal representation learning for translation-based POI recommendation. *ACM Trans. Inf. Syst.* 37, 2, Article 18 (Jan. 2019), 24 pages. DOI: https://doi.org/10.1145/3295499

[49] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09).* AUAI Press, Arlington, VA, 452–461. http://dl.acm.org/citation.cfm?id=1795114.1795167.

[50] Alan Said, Ben Fields, Brijnesh J. Jain, and Sahin Albayrak. 2013. User-centric evaluation of a K-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW'13).* ACM, New York, NY, 1399–1408. DOI: https://doi.org/10.1145/2441776.2441933

[51] Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. 2014. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14).* ACM, New York, NY, 209–216. DOI: https://doi.org/10.1145/2645710.2645743

[52] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11).* ACM, New York, NY, 109–116. DOI: https://doi.org/10.1145/2043932.2043955

[53] Saúl Vargas and Pablo Castells. 2014. Improving sales diversity by recommending users to items. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14).* ACM, New York, NY, 145–152. DOI: https://doi.org/10.1145/2645710.2645744

[54] Hao Wang, Yanmei Fu, Qinyong Wang, Hongzhi Yin, Changying Du, and Hui Xiong. 2017. A location-sentiment-aware recommender system for both home-town and out-of-town users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17).* ACM, New York, NY, 1135–1143. DOI: https://doi.org/10.1145/3097983.3098122

[55] Hao Wang, Huawei Shen, Wentao Ouyang, and Xueqi Cheng. 2018. Exploiting POI-specific geographical influence for point-of-interest recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18).* International Joint Conferences on Artificial Intelligence Organization, 3877–3883. DOI: https://doi.org/10.24963/ijcai.2018/539

[56] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web (WWW'17).* International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 391–400. DOI: https://doi.org/10.1145/3038912.3052638

[57] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. 2015. Geo-SAGE: A geographical sparse additive generative model for spatial item recommendation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15).* ACM, New York, NY, 1255–1264. DOI: https://doi.org/10.1145/2783258.2783335

[58] L. Wu, Q. Liu, E. Chen, N. J. Yuan, G. Guo, and X. Xie. 2016. Relevance meets coverage: A unified framework to generate diversified recommendations. *ACM Trans. Intell. Syst. Technol.* 7, 3, Article 39 (Feb. 2016), 39:1–39:30 pages.

[59] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based POI embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM'16).* ACM, New York, NY, 15–24. DOI: https://doi.org/10.1145/2983323.2983711

[60] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17).* ACM, New York, NY, 1245–1254. DOI: https://doi.org/10.1145/3097983.3098094

[61] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. 2013. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media (HT'13).* ACM, New York, NY, 119–128. DOI: https://doi.org/10.1145/2481492.2481505

[62] Lina Yao, Quan Z. Sheng, Yongrui Qin, Xianzhi Wang, Ali Shemshadi, and Qi He. 2015. Context-aware point-of-interest recommendation using tensor factorization with social regularization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15).* ACM, New York, NY, 1007–1010. DOI: https://doi.org/10.1145/2766462.2767794

[63] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, NY, 325–334. DOI: https://doi.org/10.1145/2009916.2009962

[64] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013. LCARS: A location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. ACM, New York, NY, 221–229. DOI: https://doi.org/10.1145/2487575.2487608

[65] Hongzhi Yin, Xiaofang Zhou, Yingxia Shao, Hao Wang, and Shazia Sadiq. 2015. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM'15)*. ACM, New York, NY, 1631–1640. DOI: https://doi.org/10.1145/2806416.2806500

[66] Haochao Ying, Liang Chen, Yuwen Xiong, and Jian Wu. 2016. PGRank: Personalized geographical ranking for point-of-interest recommendation. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW'16 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 137–138. DOI: https://doi.org/10.1145/2872518.2889378

[67] Fajie Yuan, Joemon M. Jose, Guibing Guo, Long Chen, Haitao Yu, and Rami S. Alkhawaldeh. 2016. Joint geo-spatial preference and pairwise ranking for point-of-interest recommendation. In *Proceedings of the 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI'16)*. 46–53. DOI: https://doi.org/10.1109/ICTAI.2016.0018

[68] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. ACM, New York, NY, 363–372. DOI: https://doi.org/10.1145/2484028.2484030

[69] Quan Yuan, Gao Cong, and Aixin Sun. 2014. Graph-based point-of-interest recommendation with geographical and temporal influences. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM'14)*. ACM, New York, NY, 659–668. DOI: https://doi.org/10.1145/2661829.2661983

[70] Cheng Xiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR'03)*. ACM, New York, NY, 10–17. DOI: https://doi.org/10.1145/860435.860440

[71] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: Personalized geo-social location recommendation: A kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'13)*. ACM, New York, NY, 334–343. DOI: https://doi.org/10.1145/2525314.2525339

[72] Jia-Dong Zhang and Chi-Yin Chow. 2015. 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 (SIGIR'15)*. ACM, New York, NY, 443–452. DOI: https://doi.org/10.1145/2766462.2767711

[73] Jia-Dong Zhang, Chi-Yin Chow, and Yu Zheng. 2015. ORec: An opinion-based point-of-interest recommendation framework. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM'15)*. ACM, New York, NY, 1641–1650. DOI: https://doi.org/10.1145/2806416.2806516

[74] Mi Zhang and Neil Hurley. 2009. Novel item recommendation by user profile partitioning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01 (WI-IAT'09)*. IEEE Computer Society, Washington, DC, 508–515. DOI: https://doi.org/10.1109/WI-IAT.2009.85

[75] Wei Zhang and Jianyong Wang. 2015. Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM'15)*. ACM, New York, NY, 1221–1230. DOI: https://doi.org/10.1145/2806416.2806564

[76] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R. Lyu. 2017. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW'17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 153–162. DOI: https://doi.org/10.1145/3041021.3054138

[77] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael Lyu, and Irwin King. 2016. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation.

[78] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web (WWW'05)*. ACM, New York, NY, 22–32. DOI: https://doi.org/10.1145/1060745.1060754


Received September 2018; revised May 2019; accepted August 2019