2016 17th IEEE International Conference on Mobile Data Management

MPG: Not so Random Exploration of a City

Xiaoyu Ge[1], Panos K. Chrysanthis[1], Konstantinos Pelechrinis[2]

[1] *Department of Computer Science, University of Pittsburgh, Pittsburgh, PA, USA*
{xig34,panos}@cs.pitt.edu

[2]*School of Information Sciences, University of Pittsburgh, Pittsburgh, PA, USA*
kpele@pitt.edu

***Abstract*— The proliferation of mobile, ubiquitous and spatial computing has led to a number of services aiming into facilitate the exploration of a city. Platforms such as Foursquare and Yelp curate information about establishments in an area that can then be used for recommendation purposes. Traditionally an approach followed by these systems is to rank places based on their popularity, proximity or any other feature that represents the quality of the venue and then return the top-k of them. However, this approach, while simple and intuitive, is not necessarily providing a diverse set of recommendations, since similar venues typically are ranked closely. Therefore, in this paper we design and introduce `MPG` (which stands for Mobile Personal Guide), a mobile service that provides a set of diverse venue recommendations better aligned with user preferences. `MPG` takes into consideration the user preferences (e.g., distance willing to cover, types of venues interested in exploring, etc.), the popularity of the establishments, as well as their distance from the current location of the user by combining them in a single composite score. We evaluate our approach using a large-scale dataset of approximately 14 million venues collected from Foursquare. Our results indicate that `MPG` can increase coverage of the result set compared to the baselines considered. It further achieves a significantly better *Relevancy-Diversity* trade-off ratio.**

I. Introduction

The rapid developments in mobile computing has lead to the transformation of traditional Yellow pages to mobile. Platforms such as Yelp and Foursquare allow their users to generate content (e.g., text, image, etc.) and share their experiences with their peers. This content is consequently consumed by other users, thus, closing the communication channel and allowing the exploration of an urban area.

Many systems have been developed and built on top of these platforms for recommending specific venues to be visited by users, i.e., a *digital travel guide*. Given that this digitization results in a richer and up-to-date content, the possibilities for providing a flexible, personalized guide are huge. Nevertheless, many of the approaches to date are monolithic and myopic to the user preferences, returning generic recommendations where every location is treated equally (e.g., [11], [13], [20], [28], [34]). Of course, personalized tour systems have also appeared in the literature (e.g., [16] with more details provided in Section VI) taking into consideration spatiotemporal constraints, users' interests, etc. However, a common theme among these approaches - personalized or not - is the ranking of venues based on some quality features. Consequently, the top-k venues are returned. The drawback of this approach is that it does not allow for a **diverse** set of recommendations; similar venues will tend to have similar ranking and hence the top venues will all be similar to each other with high probability. At a high level this translates to a poor recommendation since the *effective* choice of the user is reduced, given that many of the recommended venues will offer similar experiences.

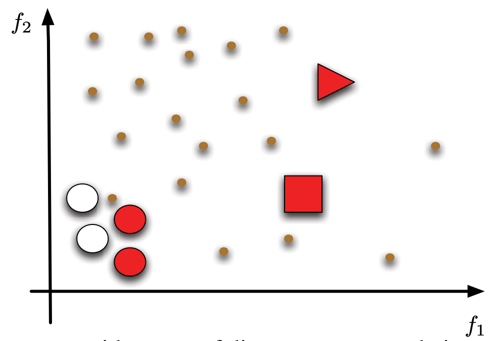

Fig. 1: `MPG` provides a set of diverse recommendations without sacrificing the quality.

In this work, our goal is to design the Mobile Personal Guide (`MPG`) that will take into consideration the user's preferences and provide a set of venues that satisfies the imposed constraints with maximized diversity. The diversity (to be formally defined later) is essentially a measure of dissimilarity of the venues based on external attributes. Simply put, `MPG` outputs a set of high-quality yet diverse venues. To illustrate this objective let us consider a toy-example consisting of 100 venues that satisfy Pam's preferences depicted in Fig. 1. Assume, Pam only has time to visit 4 of them. The venues represented by the large circles correspond to the top-4 venues ranked based on their popularity for instance. The venues represented by the triangle and the square are the 5th and 6th ranked venues, respectively. The rest of the venues are lower ranked and are represented with the brown dots. The space corresponds to two external features (f_1, f_2) that define the similarity of a pair of venues. In particular, the top-4 venues as we can see are very close in this space and hence are similar (or in other words they have low diversity in the space defined by f_1 and f_2). A system that does not consider the diversity of the recommended venues would most certainly choose these venues as the output. However, `MPG` allows the user to explore the available venues in this latent space - without sacrificing the quality of the recommendations - and so, it would return

2375-0324/16 $31.00 © 2016 IEEE
DOI 10.1109/MDM.2016.24

to the user the top-2 venues as well as the 5th and 6th ranked venues (the venues with red fill). As we can see this set is more spread in this latent space as compared to the top-4 venues.

In a nutshell our approach consists of the following basic steps. First, we begin by assigning an intensity value I_p^v to venue v based on its *popularity*. We also assign a distance intensity value $I_{d,q}^v$, which captures the distance between the current location q of the mobile user and venue v. By combining I_p^v and $I_{d,q}^v$ we obtain an updated intensity value, $I_{p,d}^v$. We then further tune these intensity scores based on the preferences of user u obtaining $I_{p,d,u}^v$, which forms our composite intensity value spaceI_k^v. Finally, I_k^v along with a vector $\mathbf{f}_v$ that represent venue v in the latent space (i.e., external attributes) form the input to our slightly modified *PrefDiv* algorithm [14] whose output is the required recommendations. One of the advantages of *PrefDiv* is that it offers to the mobile user the ability to adjust the balance between relevance and diversity in the returned results.

A main focus of our study is an appropriate definition for the popularity-based intensity value I_p^v. A straightforward approach is to consider information such as the number of total visitations in venue v and/or the number of unique people that have visited v. However, this might introduce *age biases*, that is, venues that are older will inevitably have accumulated a larger number of visitations. Furthermore, venues in a city are not isolated entities. They *interact* with each other as part of a large, connected network based on the aggregate mobility of the dwellers. For example, even though venues v_i and v_j have similar number of visitations, v_i might attract customers from a large number of other establishments, while venue v_j from only a handful of them. Similar differences and aspects of a venue's popularity can be captured by analyzing a flow network between venues. In particular, we examine the integration of Page Rank in the computation of I_p^v. Our results presented in detail later indicate that Page Rank integration offers marginal gains, if at all. Given its high computational complexity, especially in a large and densely populated area where the urban flow network is expected to be larger, our final recommendation for `MPG` is to utilize the I_p^v that does not incorporate the Page Rank scores.

In summary, this paper's **contributions** are as follows:

- We introduce a new method, which capable of generating venue recommendations that are not only *popular* and *relevant* to user's preference but are also *diverse*. Our method ranks venues based user preferences, how far the venue is, and the popularity of a venue based on check-in information. It achieves diversity using the semantic distance function called Word2Vec [24]. (Sec. III)
- As a proof-of-concept, we design and implement `MPG`, a prototype of a real mobile service that provides users with a fine control over the trade-off between relevancy and diversity through intuitive tunable parameters. (Sec. IV)
- We experimentally show that `MPG` can successfully increase coverage of the result set compared to other alternatives, and achieves a significantly better *Relevancy-Diversity* trade-off ratio than other models. (Sec. V)

II. Background

In this section we will provide the description of the dataset used as well as notations for the development of `MPG`.

A. Datasets

In our work we have used data collected from the major location-based social network, Foursquare. Foursquare is a digital social network where the main interaction among its users is the voluntary sharing of one's whereabouts through *check-ins*. Foursquare has a rich, user-curated, venue database through which users can choose to notify their friends for their current location. In particular, our study utilize the following information:

Venue database: We used Foursquare's public venue API and queried information for **14,011,045** venues. Each reading has the following tuple format: `<ID, latitude, longitude, # check-ins, # unique users, type>`. The purpose of this dataset is two-fold; (a) we obtain a database of all points-of-interest (POIs) in a city, and (b) we obtain information that can be used as a proxy for the quality of a venue (e.g., the number of unique users that have checked-in to the venue or the total number of check-ins).

Venue transition flows: Foursquare's public venue API (`NextVenues` endpoint) allows us to obtain for every Foursquare venue v, a set $\mathcal{V}_v$ of venues that users *typically* visit after v. The results are based on the number of users that have performed the transition $v \rightarrow u, u \in \mathcal{V}_v$. We have queried the Foursquare venue database and have obtained the relevant information for all the venues in New York City (NYC) and San Francisco (SF).

User check-in information: User preferences can be indirectly revealed through their historic visitations (e.g., frequent visits at Chinese restaurants by Pam is a strong signal for her appeal to this cuisine). In order to build realistic user profiles for our evaluations we used a dataset collected by Cheng et al. [8] that includes geo-tagged, user-generated content from a variety of social media between September 2010 and January 2011. This dataset includes 11,726,632 Foursquare check-ins generated by 188,450 users.

B. Relevance, Intensity, Diversity and Similarity

We now formally introduce the relevance and diversity, which are central to our work.

Relevance: We represent the degree or score of relevance of an item o to a user u by the *Preference Intensity Value* (I_u^o).

Definition 1: Preference Intensity Value A Preference Intensity Value (I) is a decimal value between -1 and 1 that is used to express a negative preference, a positive preference, or equality/indifference. Negative preferences are expressed using any value in $[-1, 0)$; -1 is used to express complete dislike. Positive preferences are expressed using any value in $(0, 1]$; 1 is used to capture the most likability. Equality/indifference is expressed using 0.

Diversity: We measure the diversity of a set of items S by measuring how dissimilar, i.e., the semantic distance beyond a threshold, each item in S is with respect to each other.

Definition 2: Dissimilarity Let O be the set of items in the database. Two objects o_i and $o_j \in O$ are dissimilar to each other $dsm_\varrho(o_i, o_j)$, if $dt(o_i, o_j) > \varrho$ for some distance function dt and a real number ϱ, where ϱ is a distance parameter, which we call radius.

Definition 3: Similarity Let O be the set of items. Two objects o_i and $o_j \in O$ are similar to each other, if $dt(o_i, o_j) \le \varrho$ for some distance function dt and a real number ϱ. We use $sim_\varrho(o_i, O)$ to denote a set of items in O that are similar to an item o_i, such that $\forall o_j \in sim_\varrho(o_i, O), o_j \neq o_i$.

C. Preferential Diversity

In this section, we present *Preferential Diversity* (*PrefDiv*) [14], which we have previously proposed as an efficient solution to the Maximum Covering Diversified Top-k problem in traditional databases. *PrefDiv* is an iterative algorithm that utilizes a ranking model that produces an initial result set of objects for a given user query and returns a set of k objects with maximized relevance and diversity. *PrefDiv* is shown in Algorithm 1 and its input parameters in Table I.

Parameter A is used to tune the balance between relevance and diversity in the returned result set. Specifically, A defines the distribution of the intensity values of objects in the final result set R. When $A = 1$, R would simply be the top k objects from the initial set, i.e., the objects with the k highest intensity values. When $A = 0$, R contains k dissimilar objects from the initial set. When A is between 0 and 1 and given that PrefDiv is an iterative algorithm, the final result will have at least $A*k$ objects from every iteration, and, in each iteration, A will be divided by half. For example, when $A = 0.5$ and $k = 20$, the first iteration will select at least $20*0.5$ items for the final result set, the second iteration will select at least $20*(0.5*0.5)$ items, and so on.

The basic logic of *PrefDiv* is as follows: It first sorts the objects in the initial set $O = \{o_1, ..., o_n\}$ in descending order along their intensity value and splits them in groups of k objects. In each iteration, it evaluates the objects in a group for diversity, starting with the first group with the highest intensity objects. The item o_i with the highest $I_u^{o_i}$ in the group T_O is moved into the final result set R, if there is no object in R similar to o_i, i.e., $sim_\varrho(o_i, R)$ is empty; otherwise it is marked as "Eliminated". Also, all objects in $sim_\varrho(o_i, T_O)$ are marked as "Eliminated". While there are still objects left in T_O that are not being marked as "Eliminated", it processes the next unmarked one o_j with the highest $I_u^{o_j}$ in the same manner. It ends an iteration by finalizing the moved objects

TABLE I: Parameters of *PrefDiv*

Par.	Range	Usage
O	$1 \le \lvert O\rvert$	Set of objects with intensity values
k	$1 \le k$	Size of the result set
ϱ	$0 \le \varrho \le M^1$	Determines whether a pair of objects are similar.
A	$0 \le$A≤ 1	Determines the number of objects to be promoted to the result set at each iteration.

[1] M = Max distance of dataset

Algorithm 1 *PrefDiv*

Require:
One set of objects O, a size k, a relevancy parameter A, and a radius ϱ
Ensure:
One subset R of O
$S \leftarrow \emptyset$
$turnCounter = 0$
while there exists unmarked items in O and $|R| < k$ **do**
    Increase $turnCounter$ by 1
    $S \leftarrow$ Pick k items with highest intensity from O
    for all items $o_i \in R$ **do**
        for all items $o_j \in S$, s.t. $o_j \in sim_r(o_i, S)$ **do**
            Mark o_j as "Eliminated"
    while there exists unmarked items in S **do**
        $R = R \cup o_i$, s.t. $o_i \in S$ is unmarked and $I_u^{o_i} \ge I_u^{o_j} : \forall o_j \in S$
        for all unmarked $o_u \in S$ **do**
            if $o_u \in sim_r(o_i, S)$ **then**
                mark o_u as "Eliminated"
    while number of unmarked items in $S < A \cdot k$ **do**
        $R = R \cup o_i$, s.t. $o_i \in S$ is unmarked and $I_u^{o_i} \ge I_u^{o_j} : \forall o_j \in S$
    $A = A \cdot 0.5$
    if $turnCounter == 1$ **then**
        create new set $G \leftarrow \forall o_j \in S$, s.t. o_j is marked
    $O = O - (O \cap S)$
if $|R| < k$ and $\forall o_j \in O$, s.t. o_j are marked **then**
    while $|R| < k$ **do**
        $R = R \cup o_j$, s.t. $o_j \in G$ and $I_u^{o_j} \ge I_u^{o_i} : \forall o_i \in G$
Return R

into R according to A, as mentioned above. If fewer than the required $A * k^{iteration}$ objects were moved in R, then the difference s is covered by moving the top-s objects with the highest intensity values that have been marked as "Eliminated" in T_O into R. The iterations continue until either k objects are selected ($|R| = k$), or if all items in O are examined. If the size of R is still less than k, $k - |R|$ items with the highest intensity values that have been marked as "Eliminated" will be selected and added into R.

PrefDiv is linear to the number of objects in the initial set. The initial candidate selection for first iteration takes $O(k^2)$ and each subsequent iteration costs $O(k^2)$ as well. As there are at most $\frac{N}{k}$ iterations, Algorithm 1 has an overall worst case complexity of $O(kN)$.

D. Venue Flow Network

As mentioned in the Introduction, a critical element in the design and evaluation of MPG is the definition of the popularity-based intensity value. Towards that end, we will examine the integration of a flow network $\mathcal{G}_f$ between venues in a city as captured through the aggregate mobility of city-dwellers. This network is derived by the venue transition flows dataset obtained from Foursquare and essentially captures the transitions of people between the venues in an area. Formally, $\mathcal{G}_f$ is defined as follows:

Definition 4: The venue flow network $\mathcal{G}_f = (\mathcal{V}, \mathcal{E})$, is a directed network where a node $v_i \in \mathcal{V}$ represents a venue and there is a directed edge $e_{ij} \in \mathcal{E}$ from node v_i to node v_j iff v_j has been visited immediately after v_i.

In particular, we study the use of the PageRank of $\mathcal{G}_f$ in the definition of a popularity-based intensity for a venue. With $\boldsymbol{\beta}$ being a vector whose i-th element captures external (i.e., irrelevant from the network structure) factors affecting the centrality of node v_i, the PageRank of $\mathcal{G}_f$ is given by [25]:

$$\boldsymbol{\pi} = D(D - \alpha A)^{-1}\boldsymbol{\beta} \tag{1}$$

where A is the adjacency matrix of $\mathcal{G}_f$, α is a parameter (a typical value of which is 0.85) and D is a diagonal matrix where $d_{ii} = \max(1, k_{i,out})$, with $k_{i,out}$ being the out-degree of node v_i. While in most practical cases PageRank considers only the network structure, Eq. (1) is able to take into consideration - if needed - not only the network structure but external information that affect the "importance" of a node/venue v_i through vector $\boldsymbol{\beta}$. In MPG we do not consider this option and hence, $\boldsymbol{\beta}$ is the unit vector.

III. System Design

In this section we will begin by formally presenting the mobile personal guide problem and consequently detailing the design of MPG.

A. Problem Statement

We begin by formalizing the algorithmic problem that lays in the epicenter of MPG.

*Problem 1 (*MPG*):* Given a set of geographical points $V = \{v_1, v_2, \ldots, v_l\}$, a popularity index ξ_{v_i} for location v_i, a query point q, a reach r, and a profile set that encodes user preferences $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, identify a set $V^* \subseteq V$ ($|V^*| = k$) with maximized diversity $\Delta(\mathcal{S})$, while a set of constraints $h(V^*, \mathcal{P}, q, r, \boldsymbol{\xi})$ is satisfied.

In our setting the set V corresponds to the set of available venues/Points-of-Interest. The query point q corresponds to the current location of the mobile user, while r represents the maximum allowed distance between q and any point in the chosen solution V^*. The set of preferences $\mathcal{P}$ captures the profile of the mobile user with respect to his interests. In our setting, we will use the user check-in information dataset (see Section II) from Foursquare to build the users' profiles as we detailed in Section III-B. Finally, the constraints described by function h in Problem 1 include, (i) a geographic constraint that ensures that the maximum distance between the currently location of the mobile user and any venue recommended does not exceed r (i.e., $d(q, v_i) \leq r, \forall v_i \in V$), and (ii) a personalization constraint that ensures that the output set of venues is compatible with the user preferences (i.e., $V \models \mathcal{P}$).

Given this problem setting, the actual mobile personal guide system will include an interface that (a) will obtain the current location of the user q (e.g., through the GPS sensors, NFC

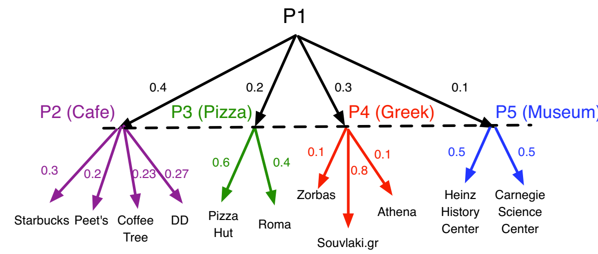

Fig. 2: Pam's sample hierarchical user profile. The first level corresponds to the coarse-grain preference profile (P_1), while each one of the sub-trees stemming from P_1 corresponds to the preferences within each category (e.g., preference P_2 corresponds to the "Cafe" venue type).

sensors etc.) and will allow the user to provide as input (b) the reach r, (c) the set of types of venues she is interested in this trip and (d) the number of venues k she would like to know about. The preference of the user will be "hardcoded" either in the system (i.e., bound with the user account) or stored on the mobile device and uploaded at the time of the request. MPG will finally provide the set V^* of the recommended venues based on the definition of Problem 1.

B. User Preferences

In Foursquare every individual venue v is associated with a type $\mathcal{T}_v$. This classification is hierarchical, in the sense that an Italian restaurant belongs to the category "Italian restaurant", which can belong to the higher level category "Restaurants", which can itself belong to the category "Food" and so on. At the top level of the hierarchy there are ten categories, namely, *Arts & Entertainment*, *College & University*, *Food*, *Nightlife Spots*, *Outdoors & Recreation*, *Events*, *Professional & Other Places*, *Residences*, *Shops & Services* and *Travel & Transport*. However, in order to build highly personalized and specific profiles we use the bottom layer of hierarchy as well as the specific venues visited.

In particular, given the set of check-ins $\mathcal{C}_u$ of mobile user u, we provide a hierarchical profile $\mathcal{P}$. At the top level, the preferences of the user are expressed in terms of the (normalized) frequencies of this user's visitations with respect to the types of venues. The second layer of the user profiles further provides the normalized frequencies of venues for the different types of locations visited by u. Fig.2 presents a sample profile for Pam. Preference P_1 is a coarse-grain preference profile, which informs the system that Pam prefers to spend 40% of her time in coffee shops, 10% in museums, 20% in burger joints and 30% in Greek restaurants. Preferences $P_2 - P_5$ are able to distill further Pam's preferences. For instance, she appears to prefer Starbucks more compared to Peet's coffee.

C. Distance-based Intensity Value

The physical distance between the current location q of the mobile user and venue v can also be used to obtain an intensity value for v. In particular with d_q^v being the normalized distance between q and v's location the distance-based intensity value can be defined as:

$$I_d^v = 1 - \frac{d_q^v}{r} \tag{2}$$

In the above equation the distance has been normalized based on the maximum allowed distance from Problem 1, that is, r. Note here that d_q^v can be, in principle, equal to 0. However, this happens when the current user location q is at venue v. Given that the user is already at this location these venues are not considered by our system.

D. Popularity-based Intensity Value

An important factor that can impact the choice of a venue v from MPG is its popularity. With c_v being the number of total visits in venue v, i.e., the number of check-ins in v, and s_v being the number of unique visitors in v, we define the popularity-based intensity value of v as:

$$I_p^v = \lambda \cdot \frac{c_v}{\max\limits_{i \in V} c_i} + (1-\lambda) \cdot \frac{s_v}{\max\limits_{i \in V} s_i}, \ \lambda \in [0,1] \tag{3}$$

where V is the set of all the venues. This intensity value essentially corresponds to the popularity index $\boldsymbol{\xi}$ used in the formal definition of the MPG problem.

Eq. 3 does not consider the flow network between venues that can provide additional popularity information for venues. Hence, with π_v being the Page Rank score for venue v, we updated Eq. 3 as follows:

$$I_{p,\pi}^v = \mu \cdot (\lambda \cdot \frac{c_v}{\max\limits_{i \in V} c_i} + (1-\lambda) \cdot \frac{s_v}{\max\limits_{i \in V} s_i}) + (1-\mu) \cdot \pi_v, \ \mu, \lambda \in [0,1] \tag{4}$$

Having I_d^v and I_p^v (or $I_{p,\pi}^v$), we can combine them in one intensity score as follows:

$$I_{d,p}^v = \gamma \cdot I_d^v + (1-\gamma) \cdot I_p^v, \ \gamma \in [0,1] \tag{5}$$

E. Preference-based Intensity Value

The degree or strength of relevance of a venue v is expressed by the preference-based intensity value I_u^v derived from the user's profile. In particular, the preference-based intensity value is a combination of the score of the type of the venue (i.e., the coarse-grain preference score) with the specific venue (i.e., fine-grain preference) score. As stated above, since these scores are derived from the user's check-ins C_u, the preference-based intensity value I_u^v for venue v and user u is computed as follows:

$$I_u^v = 0.5 \cdot \frac{C_u^v}{\sum\limits_{v_j \in t} C_u^{v_j}} + 0.5 \cdot \frac{\sum\limits_{v_j \in t} C_u^{v_j}}{\sum\limits_{t \in T} \sum\limits_{v_j \in t} C_u^{v_j}} \tag{6}$$

where, C_u^v is the number of check-ins that u had in v, t is the venue type of v and T is the set of all venue types.

We can further combine I_u^v with $I_{d,p}^v$ in a manner similar to Equation (5) and obtain a value that combines the user preference, the popularity (with or without the integration of Page Rank) and the distance of the venue from the current location of the user. More specifically:

$$I_{u,p,d}^v = \alpha \cdot I_u^v + (1-\alpha) \cdot I_{d,p}^v, \ \alpha \in [0,1] \tag{7}$$

F. Composite Intensity Value

Eq. (7) combines 3 different elements (user preference through I_u^v, venue popularity through I_p^v or $I_{p,\pi}^v$ and geography through I_d^v) into a single intensity score. This combined intensity score is the composite intensity value of v, I_k^v (or $I_{k,\pi}^v$ if Page Rank is used in the popularity intensity value). One point we would like to emphasize here is that the order with which we combine the three intensity values (i.e., I_u^v I_p^v (or $I_{p,\pi}^v$) and I_d^v) to obtain I_k^v (or $I_{k,\pi}^v$) does **not** impact the output of MPG. The reason is that MPG outputs a total order of the venues based on these three factors. The absolute values themselves for I_k^v will be different, but the order will always be the same.

IV. MPG Prototype Implementation

The MPG prototype essentially implements the *PrefDiv* algorithm (Section II-C) with a parameterized intensity value and Word2Vec [24], as the semantic distance function. For efficiency, the implementation makes extensive use of hash tables and indexes. The two key indexes used are the *M-Tree* [9] and the *Category Tree*, which are described below along with our Word2Vec implementation.

A. M-Tree

One of the main operations in MPG is to generate a set of nearby neighbors. In order to speed up this process, MPG utilizes the well-known *M-tree* spatial index structure [9]. M-tree uses triangle inequality for efficient range queries similar to those required in MPG. An M-tree is a balanced tree index that is designed to handle a large scope of multi-dimensional dynamic data in general metric spaces. An M-tree partitions the space in such way that it generates bounding ball regions around some of the indexed items, called *pivots*, with some bounding radius r. Each internal node has at most N entries, and contains the following attributes: a pivot p_v, the bounding radius r around p_v, a pointer pt to the subtree that is rooted at the pivot p_v, and the distance between p_v and its parents pivot. The distance of a subtree from p_v is guaranteed to be within the bounding radius r. Each leaf node in the tree will have two attributes: the item that is being indexed, and the distance between this leaf node and the parent pivot. In MPG, we have modified the implementation of the M-Tree from [23].

B. Category Tree

MPG uses the Foursquare Category Hierarchy [1] first to derive the user preferences and build user profiles, and second in the comparisons for similarity among venues. MPG accelerates both of these operations by building a *category tree* to capture the category structure of venues in Foursquare as a tree. Each internal node in the category tree represents a type of venue, where each internal node represents the subcategory of the parent node with each leaf node representing the actual venue. There are in total 10 categories at the top-level of this hierarchy. Each internal node in a category tree contains the following attributes: ID of the category it represents, name of the category, a pointer to the parent node and a list of pointers to each of its children nodes. Since a category tree can have an unlimited number of degrees, all the children node pointers are stored as hash tables, with the key being venue ID and the value being the actual pointer.

The user profiles are further derived from the preference hierarchy, as described above in Section III-B. The preference hierarchy consists of the top-level categories and the leaf nodes of the category tree (Fig. 2).

The category tree can be used to calculate the similarity distance between two venues v_i and v_j as follows:

$$Sim_{Tree}(v_i, v_j) = 1 - \frac{Ancestors\_Path}{Longest\_Path} \tag{8}$$

where *Ancestors_Path* is the number of common ancestors between the venues v_i and v_j and *Longest_Path* is the number of nodes on the longest path to the root from either v_i and v_j.

C. Word2Vec

Although the category tree is able to measure the similarity between two venues, this measurement is not very accurate as it only provides a coarse granularity distance between two venues. Specifically, this measurement cannot distinguish the difference between two venues that are under the same subcategory, for example, “McDonald’s” and “Burger King”, as both of them share the exact same ancestors.

In order to overcome this limitation, MPG also utilizes Word2Vec [24], an advanced NLP technique, which supports fine granularity distance calculation between two venues by going beyond syntactic comparisons.

Word2Vec is a tool that provides the implementation of two word vector representation computing models: *Continuous Bag-of-Words* model (CBOW), which predicts the current word based on the sourcing words, and *Continuous Skip-gram* model, which seeks to use the current words to predict surrounding words. Both of these models are based on the Neural Net Language Model. With Word2Vec, the similarity of word representations goes beyond simple syntactic regularities. Specifically, word vectors capture many linguistic regularities. For example, after obtaining the word representation in vector space, the resulting vector can have the following properties, such that vector(‘King’) - vector(‘Man’) + vector(‘Woman’) results in a vector that is closest to the vector(‘Queen’). MPG uses CBOW model to generate all word vectors.

The difference between two words under Word2Vec are calculated through the cosine similarity of two-word vectors, such that cosine similarity is defined as following:

$$Sim_{Vec}(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}} \tag{9}$$

where n is the length of vector, A_i and B_i are elements of vector A and B, respectively.

The current word vectors we adopted support phrases that consist of up to two words. For venue names that have more than two words or are not contained in the word vectors, we split the phrases into single words and then obtain word vectors for each individual word in the phrases. The final vector of a phrase is obtained through the average of all vectors for each word in this phrase. Since the accuracy of Word2Vec is strongly depends on the quality of the word vectors, a large real-world corpus is needed in order to obtain high quality word vectors. We have experimented with various corpus in an attempt to generate the highest quality word vectors. The best suitable word vectors we obtained were generated from the entire English wikipedia that consist of 55 GB of plain text. The resulting word vectors contain over 4 million entries. In order to effectively query the word vectors, MPG stores all the word vectors in memory as a hash map.

Similar to category-tree based similarity, the Word2Vec based similarity has its own biases. We were able to overcome these biases of the individual similarity metrics by combining them (Eqs. 8 and 9) and measuring the similarity between two venues v_i and v_j as follows:

$$Sim(v_i, v_j) = \frac{Sim_{Tree}(v_i, v_j) + Sim_{Vec}(A, B)}{2} \tag{10}$$

where A and B are representing the vector representation of venue v_i and v_j respectively.

V. Performance Evaluation

In order to study the effectiveness of MPG, we use as baseline *Page Rank* the original *PrefDiv* that considers only user preferences – PrefDiv was experimentally shown that it can successfully increase coverage of the result set compared to the state-of-the-art diversified top-k algorithms, and achieves a significantly better Relevancy-Diversity trade-off ratio than these algorithms [14]. In order to get a better insight into the impact of each component of the composite intensity value, we compare MPG to *PrefDiv* with different intensity value combinations. Table II summarizes all models employed in our experiments, and Table III summarizes the values of the parameters used.

We ran all our experiments on an Intel machine with Core i7 2.5Ghz CPUs, 16GB Memory and 512GB SSD and used the Foursquare datasets described in Section II-A. We created individual Foursquare user profiles as described above and three super-user profiles with more fine-grained preferences by merging the profiles of (i) 1000 Foursquare users (Super-user A), (ii) 500 Foursquare users (Super-user B), and (iii) 350 Foursquare users (Super-user C).

A. Evaluation Metrics

In our experimental evaluation, we used three well-known metrics: *Normalized Relevance* [30], *Average Similarity Distance*, and *Coverage* [12] .

Definition 5: Normalized Relevance. Let O be a set of venues and $O_k^* \subseteq O$ such that $|O_k^*| = k$. The Normalized Relevance of O_k^* is defined as the total relevance score of O_k^* over the sum of top-k highest relevance scores of O.

In our experiments, Normalized Relevance is measured in terms of composite intensity value and preference-based intensity value (i.e., original *PrefDiv*).

TABLE II: MODEL ABBREVIATION

Models	Description
PD(pref)	Uses preference-based intensity value as the relevance score for PrefDiv.
PD(pop+dist)	Uses popularity and distance from the user current location as the relevance score for PrefDiv.
PD(pref+dist+pr)	Uses preference-based intensity value, distance and PageRank as the relevance score for PrefDiv.
PD(dist+pref)	Uses preference-based intensity value and distance as the relevance score for PrefDiv.
PD(composite+PageRank)	Uses composite intensity value and PageRank as the relevance score for PrefDiv.
PD(composite)	Uses composite intensity value as the relevance score for PrefDiv.
PageRank	Only uses the result of PageRank as the final ranking without using PrefDiv.

Definition 6: Average Similarity Distance Let O be a set of venues, the average similarity distance of O represents the average of the pairwise distances of the venues in O.

In our experiments, Average Similarity Distance (ASD) is normalized to take into consideration that different methods may return as a result a list of venues with duplicates rather than a set and expressed as Redundancy Normalized Pairwise Distance (RNPD):

$$RNPD(L) = (1 - \frac{|Unique(L)|}{|L|}) * ASD \qquad (11)$$

where $Unique(O)$ represents O with out duplicates.

Definition 7: Coverage Let O be a set of venues, $O_k^* \subseteq O$ such that $|O_k^*| = k$ and $S \subseteq O$ be defined as the union of $sim_\varrho(v_i, O)$ for all $v_i \in O_k^*$. The coverage of a subset O_k^* is defined as the percentage of venues in S over the total number of venues in O, i.e., $|S|/|O|$.

B. Experimental Results

In this section, we present the evaluation of all models in our experiments. First, we compare all of them in terms of the metrics mentioned above, and then we focus on the two-best performing ones for further understanding their performance.

1) All models: We performed our experiments using the parameters in Table III randomly selecting 15 query points in each of New York City (NYC) and San Francisco (SF). Our experiments showed that the actual location of the query point (at least among those randomly selected) does not significantly impact our results. Due to space limitations, we only present the results from 15 random locations in each of the cities for Super-user A and for 10 categories (Figs. 3-10).

As we can see from Figs. 3 and 7, PD(composite) and PD(composite+PageRank) have the best performance in terms of Normalized Composite Intensity Value. In particular, they provide a 10% improvement on average. Page Rank consistently performs the worst with respect to relevance as expected, since it is oblivious to user preferences and venue popularity. In PD(composite) when PrefDiv selects the representative venues, it considers the Normalized Composite Intensity Value as the relevance criteria. This allows PrefDiv to optimize towards the Normalized Composite Intensity Value. In the case of PD(composite+PageRank), although it additionally considers the PageRank in the computation of the Normalized Composite Intensity Value, it does not seem to have a significant impact in terms of relevance.

TABLE III: PARAMETER CONFIGURATION

Parameters	Value
λ	0.5
μ	0.7
γ	0.7
α	0.5
A	0.6
K	10 - 50
Radius	1.5 km
Number of Locations	15, 50
Number of Categories of POIs Selected	5, 10

With respect to the Normalized Preference-based Intensity Value (Figs. 4 & 8) PD(pref) delivers the best performance. PD(pref) uses the Preference-based Intensity Value directly as the criteria for relevance. However, both PD(composite) and PD(composite+PageRank) still outperform the rest of the models by a significant margin, which shows that they can both satisfactorily reflect user preferences.

Figs. 5 and 9 further illustrate the performance of redundancy normalized pairwise distance of all models. Page Rank performs the best in this case. PD(composite) and PD(composite+PageRank) perform very close to the second best (i.e., PD(pref+dist+pr), PD(pop+dist)) by a small margin, with the performance difference being consistently between 3% and 5%. Moreover, Figs. 6 and 10 present the performance of all models with respect to coverage. As we can see, most of the models perform identical to each other, with PD(composite) and PD(composite+PageRank) performing slightly better.

Finally, Figs. 11 and 12 present two scatter plots that capture the trade-off between relevance and diversity. Each point in these two figures correspond to the average over 15 different locations of one super-user and one output size. Models located in the left upper corner of the figure exhibit the best diversity result, while the ones located in the lower right corner have the highest relevance scores. As we can observe, both PD(composite) and PD(composite+PageRank) are located towards the upper right corner (circled), which indicates that both PD(composite) and PD(composite+PageRank) are better able to handle the trade-off between relevance and diversity. In conclusions, these results indicate that both PD(composite) and PD(composite+PageRank) have the ability to achieve a good balance between diversity and relevance.

2) PD(composite) vs PD(composite+PageRank): As illustrated by Figs. 11 and 12 PD(composite) and PD(composite+PageRank) outperform the rest of the models

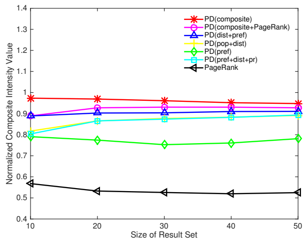

Fig. 3: Relevance: NCI (SF, Super-user A).

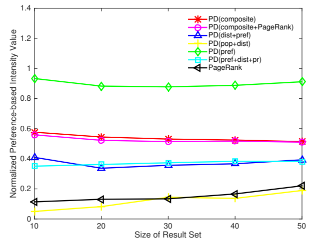

Fig. 4: Relevance: NPI (SF, Super-user A).

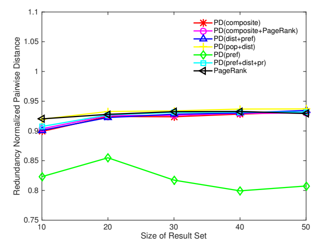

Fig. 5: Diversity: RNPD (SF, Super-user A).

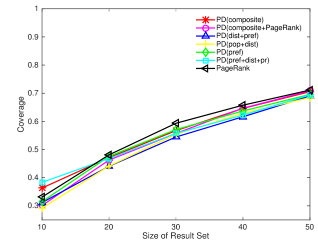

Fig. 6: Diversity: Coverage (SF, Super-user A).

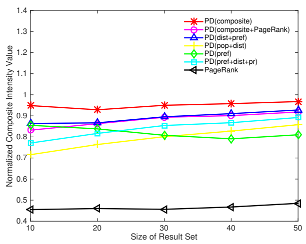

Fig. 7: Relevance: NCI (NYC, Super-user A).

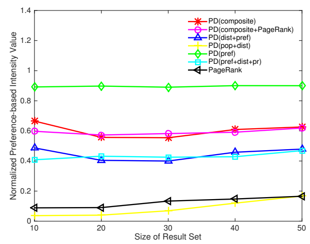

Fig. 8: Relevance: NPI (NYC, Super-user A).

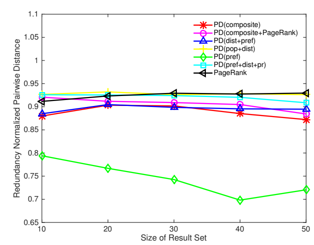

Fig. 9: Diversity: RNPD (NYC, Super-user A).

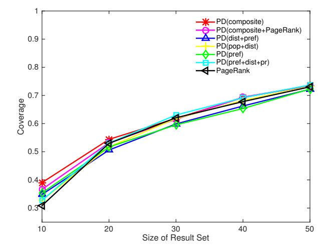

Fig. 10: Diversity: Coverage (NYC, Super-user A).

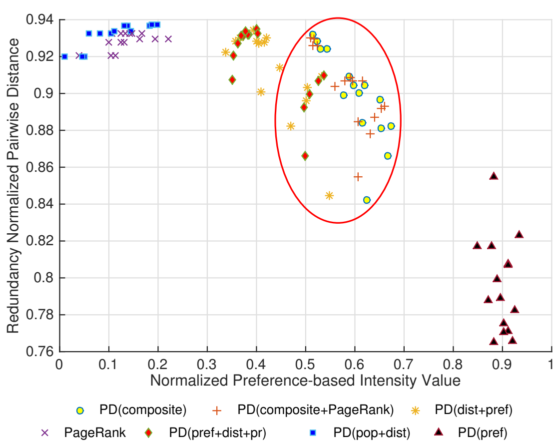

Fig. 11: Relevance VS. Diversity (SF, All Super-users).

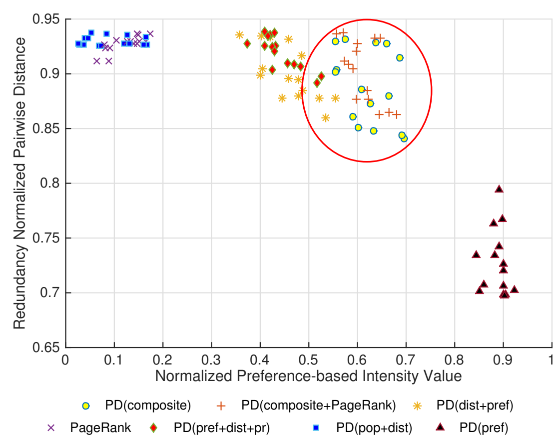

Fig. 12: Relevance VS. Diversity (NYC, All Super-users).

in balancing relevance and diversity. Thus, we carried out further experiments with these two models utilizing more query points, i.e., 50 randomly selected points in each city.

Figs. 13 and 14 demonstrates that PD(composite) outperforms PD(composite+PageRank) slightly in terms of average Normalized Composite Intensity Value. In particular, PD(composite) has a slight advantage over PD(composite+PageRank) when K is small, with the performance gap closing when the size of the output required increases. Furthermore, Figs. 15 and 16 show that both PD(composite) and PD(composite+PageRank) perform similarly in terms of Normalized Preference-based Intensity Value. The same is true with respect to the Redundancy Normalized Pairwise Distance (Figs. 17 & 18). This reflects that both models have the ability to eliminate redundant items. Finally, as shown in Figs. 19 and 20 both models provide a similar performance with respect to coverage with PD(composite) providing some marginal benefits for small K.

3) Discussion: Based on our results the performance of PD(composite) and PD(composite+PageRank) are similar, i.e., the performance of `MPG` does not significantly improve with the integration of PageRank. Given the computational complexity of PageRank, which requires $O(m+n)$ for each iteration, where n is the number of nodes and m is the number of edges in the (urban flow) network the cost of integrating Page Rank into `MPG` is high. Hence, our final design of `MPG` includes PD(composite), which returns a single recommendation in our experiments between 200 and 500 msec depending on K and the user's location.

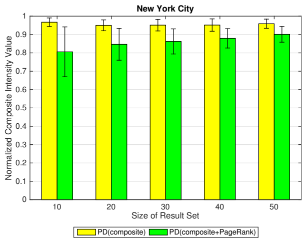

Fig. 13: Relevance: NCI (NYC, Super-user A).

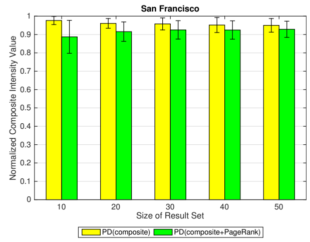

Fig. 14: Relevance: NCI (SF, Super-user A).

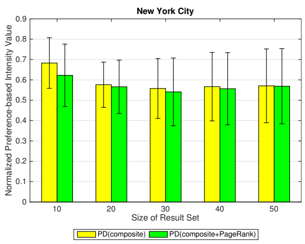

Fig. 15: Relevance: NPI (NYC, Super-user A).

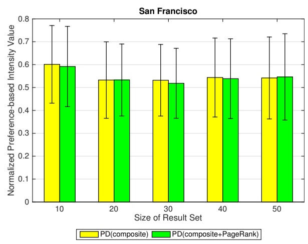

Fig. 16: Relevance: NPI (SF, Super-user A).

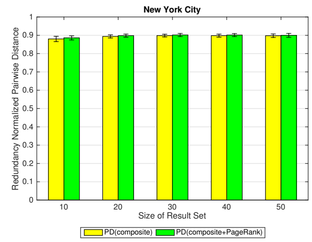

Fig. 17: Diversity: RNPD (NYC, Super-user A).

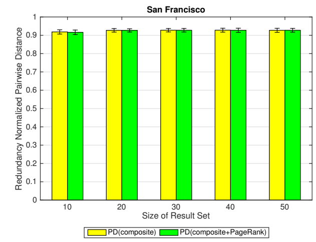

Fig. 18: Diversity: RNPD (SF, Super-user A).

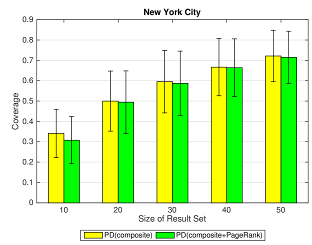

Fig. 19: Diversity: Coverage (NYC, Super-user A).

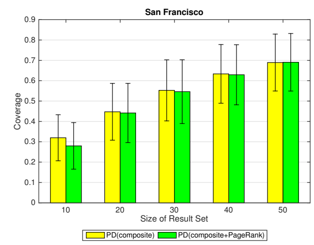

Fig. 20: Diversity: Coverage (SF, Super-user A).

VI. Related Work

In this section we will briefly discuss related to our work studies. In particular, we will present studies related with trip planning as well as with the methods of query personalization.

Trip planning and spatial recommendations: During the last years there has been a large volume of studies that focus on methods for personalized location/Point-of-Interest recommendations [4], [26], [33] to social-network users. The majority of existing work utilizes collaborative-filtering techniques [33], geometric embeddings [4] or they even incorporate features present in the users' social network [26] to associate every venue with a score, which is representative of the probability of a user enjoying (or liking) a particular venue.

Nevertheless, similar studies consider and evaluate each venue independently. Hence, motivated by this monolithic view of the above methods, recent work has focused on recommending *tours* of locations. For instance, De Choudhury et al. [11] focus on segmenting streams of spatiotemporally tagged photos into paths, and then assembling these paths into itineraries. Similar studies by Kurashima et al. [21] and Yoon et al. [34], are based on geo-tagged content from photo-sharing media (e.g., photo streams, GPS trajectories) to recommend future travel paths. However, these approaches do not come without their own drawbacks. For example, in order to be applicable, the presence of training sequences of spatiotemporally tagged photographs (or other similar traces) is required. These approaches cannot handle multiple types of venues that cater to different user needs. The same is true for interactive systems [13], [20], [28], which iteratively personalize or improve a tour based on user feedback.

The support for multiple types of venues is considered by Ardissono et al. [3] where the user *manually* selects a venue from each desired type and then a tour traversing the selected venues is proposed. More recently, Gionis et al. [16] developed a system based on dynamic programming algorithms to provide spatial-constrained tours based on users preferences of type of venues. Other spatial recommendation approaches focus on reconstructing and recommending routes based on existing location trajectories (e.g., [7], [32]).

To the best of our knowledge there is no study to date that considers the venue diversity (in a latent space) when it comes to recommendations. On a different direction though Lu et al. [22] propose the integration of the output of several location recommender systems, whose outcome are different.

Query personalization: Relevance ranking using preferences and result diversification techniques have been proposed to deal with the problem of information overload, i.e., avoid overwhelming the users with a large volume of irrelevant results. Ranking techniques are comprehensively surveyed in Stefanidis et al. [29]. Mostly these techniques can handle only one type of preference, either *quantitative* preferences or *qualitative* preferences. Hybrid schemes that support both qualitative and quantitative preferences have been proposed in an attempt to exploit the advantages of both types of preferences while eliminating their disadvantages [15], [19].

Diversity has various definitions in the literature [23]. The most common definitions are based on *similarity*, where diversity means to include in the results objects that are dissimilar to each other (e.g., [37]). Other definitions are based on either *semantic coverage*, where diversity means to include objects that belong to different categories (e.g., [2]), or *novelty*, where diversity means to include data that contains new information (i.e., information that has not been presented previously) (e.g., [10]). During the past, many result diversification models have been proposed, e.g., *MaxMin* and *MaxSum* (e.g., [5], [17], [31]) and *DisC Diversity* [12], [23].

Even though the goal of diversity is to ensure potentially important data is not lost due to its low ranking, however the result of diversification does not automatically imply relevancy for the users. This was the underlying motivation for top-k diversification techniques, such as *PrefDiv* [14], *Swap* [35] and *Div-Astar* [27], the *Query Manifold* (*QM*) framework [36] and the multi-objective optimization approaches, where the first objective is relevance and the second objective is dissimilarity [37]. As opposed to *PrefDiv* and the multi-objective optimizations that recommend relevant and diverse data, *QM* recommends a set of relevant and diverse queries.

The difference between *PrefDiv* and Swap is that Swap seeks diversity through pairwise distances of items among the result set and filters out items that contribute less to diversity. Swap ensures relevance by removing items that drop the relevance below the pre-defined threshold. In contrast, *PrefDiv* seeks diversity through eliminating similar items and ensures relevance by using a relevance-focused greedy algorithm that can reflect the user specified relevance distribution.

Div-Astar [27] is a graph-based solution in which each node corresponds to one item in the original data. This diversity graph is sorted according to the relevance score and an a^* algorithm is used to find the exact solution for diversifying top-k results. That is, *Div-Astar* considers the problem as finding the the optimal solution for the maximum weight independent set problem, which has been proven to be NP-hard.

The most widely known approach that is targeted directly at optimizing the trade-off between diversity and relevance, was introduced by [6]. In this work, the authors have purposed the twin-objective function called *Maximal Marginal Relevance* (MMR), which combines both relevance and diversity aspects in a single comprehensive objective function with a scaling factor λ. When $\lambda = 1$, the MMR function equals to a standard relevance ranking function, when $\lambda = 0$ it computes a maximal diversity ranking. Recently, a new MMR function that integrates *regret minimization* was proposed to generate the relevance score [18]. This new score attempts to minimize the disappointment of users when they see k representative tuples rather than the whole database.

VII. Conclusions

In this paper we propose and design MPG, a mobile service that provides a set of diverse venue recommendations better aligned with user preferences. This is achieved by considering the user habits, the reach willing to cover, the types of venues interested in exploring, and the popularity and the diversity of venues. Our evaluation with Foursquare data indicates that integrating Page Rank with popularity provides only marginal benefits, and hence, in view of its high computational complexity, we recommend not including it in MPG.

References

[1] Foursquare category hierarchy. `https://developer.foursquare.com/categorytree`, 2015.
[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, 2009.
[3] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence*, 17(8-9), 2003.
[4] B. Berjani and T. Strufe. A recommendation system for spots in location-based online social networks. In *SNS*, 2011.
[5] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS*, 2012.
[6] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.
[7] Z. Chen, H. Chen, X. Zhou, Y. Zheng, and X. Xie. Searching trajectories by locations: An efficiency study. In *ACM SIGMOD*, 2010.
[8] Z. Cheng, J. Caverlee, K. Lee, and D. Sui. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.
[9] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, 1997.
[10] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Bttche, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, 2008.
[11] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In *HT*, 2010.
[12] M. Drosou and E. Pitoura. Result diversification based on dissimilarity and coverage. In *VLDB*, 2012.
[13] S. Dunstall, M. Horn, P. Kilby, M. Krishnamoorthy, B. Owens, D. Sier, and S. Thièbaux. An automated itinerary planning system for holiday travel. In *Journal of IT and Tourism*, 2003.
[14] X. Ge, P. K. Chrysanthis, and A. Labrinidis. Preferential diversity. In *ExploreDB*, 2015.
[15] R. Gheorghiu, A. Labrinidis, and P. K. Chrysanthis. A user-friendly framework for database preferences. In *CollaborativeCom*, 2014.
[16] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi. Customized tour recommendations in urban areas. In *ACM WSDM*, 2014.
[17] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, 2009.
[18] Z. Hussain, H. A. Khan, and M. A. Sharaf. Diversifying with few regrets, but too few to mention. In *ExploreDB*, 2015.
[19] W. Kiessling and G. Köstler. Preference SQL: design, implementation, experiences. In *VLDB*, 2002.
[20] K. Kim, H. Kim, and J. Ryu. Triptip: a trip planning service with tag-based recommendation. In *CHI (Extended abstract)*, 2009.
[21] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In *CIKM*, 2010.
[22] Z. Lu, H. Wang, N. Mamoulis, W. Tu, and D. Cheung. Personalized location recommendation by aggregating multiple recommenders in diversity. In *ACM LocalRec*, 2015.
[23] E. P. Marina Drosou. Multiple radii disc diversity: Result diversification based on dissimilarity and coverage. *ACM TODS*, 40(1), 2015.
[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.
[25] M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.
[26] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *SocialCom*, 2012.
[27] L. Qin, J. X. Yu, and L. Chang. Diversifying top-k results. In *VLDB*, 2012.
[28] S. Roy, G. Das, S. Amer-Yahia, and C. Yu. Interactive itinerary planning. In *ICDE*, 2011.
[29] K. Stefanidis, G. Koutrika, and E. Pitoura. A survey on representation, composition and application of preferences in database systems. *ACM TODS*, 36(19), 2011.
[30] H. Tong, J. He, Z. Wen, R. Konuru, and C.-Y. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *KDD*, 2011.
[31] M. R. Vieira, H. L. Razente, M. C. N. B. M. Hadjieleftheriou, D. Srivastava, C. Traina, and V. J. Tsotras. On query result diversification. In *ICDE*, 2011.
[32] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing popular routes from uncertain trajectories. In *ACM KDD*, 2012.
[33] M. Ye, Y. Peifeng, and W.-C. Lee. Location recommendation for location-based social networks. In *ACM SIGPATIAL GIS*, 2010.
[34] H. Yoon, Y. Zheng, X. Xie, and W. Woo. Social itinerary recommendation from user-generated digital trails. In *Personal and Ubiquitus Computing*, 2012.
[35] C. Yu, L. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: Diversification in recommender systems. In *EDBT*, 2009.
[36] X. Zhu, J. Guo, X. Cheng, P. Du, and H. Shen. A unified framework for recommending diverse and relevant queries. In *WWW*, 2011.
[37] C.-N. Ziegler, J. A. K. Sean M. McNee, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.