2017 IEEE 37th International Conference on Distributed Computing Systems

Data-driven Serendipity Navigation in Urban Places

Xiaoyu Ge[1], Ameya Daphalapurkar[2], Manali Shimpi[2], Darpun Kohli[2]
Konstantinos Pelechrinis[3], Panos K. Chrysanthis[1] and Demetrios Zeinalipour-Yazti[4]
[1,2] Department of Computer Science, [3] School of Information Science, University of Pittsburgh, USA
[4] Max Planck Institute for Informatics, Germany and University of Cyprus, Cyprus
[1] {xiaoyu, panos}@cs.pitt.edu, [2] {abd51, mcs116, dak190}@pitt.edu, [3] kpele@pitt.edu, [4] dzeinali@mpi-inf.mpg.de

***Abstract*—With the proliferation of mobile computing and the ability to collect detailed data for the urban environment a number of systems that aim at providing Points of Interest (POIs) and tour recommendations have appeared. The overwhelming majority of these systems aims at providing an optimal recommendation, where optimality refers to objectives of minimizing the distance to be covered or maximizing the quality of the POIs recommended. A major problem is that by focusing on the optimization of these objectives, there remains little room to the user for *serendipity*. Urban and social scientists have identified serendipity, i.e., the ability to come across unexpected places, as a feature that makes a city livable. In this work, we introduce a prototype of an experimental platform for evaluating venue recommendation algorithms by providing informative tour recommendations based on the suggested venues. Our prototype system integrates the notion of serendipity in urban navigation at both the venue as well as the route recommendation level without compromising the quality and diversity of the recommended POIs. In addition, our system allows the user to upload their own algorithms and explore their performance as compared to many well-known algorithms.**

I. Introduction

The rapid proliferation of spatial and mobile computing has allowed people to effectively navigate through a city. While GPS technology and route-planning services that have appeared during the past years have certainly helped towards more efficient navigation in a city, they are at the same time *overly optimized*. These systems are overwhelmingly focused on optimizing an efficiency objective, such as minimizing the distance covered, maximizing the benefit obtained from the route as captured by a measure of venue quality and so on (e.g., [6], [7], [12]). It is only in recent years that efforts have been made to consider objectives that go beyond the pure efficiency (e.g., [11], [2]) but they are still in a nascent stage. While optimizing efficiency can make an urban environment acceptable it does not necessarily make for a great and livable city. What is currently missing in these systems is integrating features that actually contribute to a livable environment.

Serendipity has been identified as a characteristic that can significantly improve the quality of experience that a city-dweller has [8]. In this work we developed an experimental platform for urban navigation in order to explore our ideas for capturing serendipity and compare them with existing approaches. This led to the development of two novel schemes for capturing serendipity, both of which explore randomness. Our first scheme achieves serendipity through the recommendation of venues based on our previous proposed algorithm *Preferential Diversity* (*PrefDiv*) [3]. We introduce a probability-based variant of *PrefDiv*, namely, *Probabilistic Preferential Diversity* (*pPrefDiv*), which incorporates the serendipity while still preserving the efficiency of objectives as *PrefDiv*. Our second scheme achieves serendipity through the recommendation of routes. To do so, we utilize random walks to generate a set of initial routes. The randomness of this process essentially incorporates the required serendipity, since the venues to be included in the route are not chosen in a way that optimizes a pre-defined objective. Later, a Pareto front is deployed to pick final recommendations that show high quality with respect to both diversity and serendipity.

Our experimental platform supports a number of different approaches for the venue and tour recommendations based on the user's current location. It enables the user to compare the recommendations of different recommendation engines by both presenting the recommended venues and tours visually on the map as well as displaying evaluation metrics through summary dashboards. In addition to our methods, our platform implements the following known algorithms: DisC Diversity [9], K-Medoid and a PageRank-based recommendation approaches. In addition, our prototype system provides an interface for advanced users (e.g., researchers) to design and integrate their own recommendation algorithms into our prototype system. While we have implemented the same diversity scheme for all the algorithms, our implementation is flexible and allows for different diversity and indexing schemes that can easily be adjusted by the user.

In the next section, Section II, we present our prototype experimental platform and our new model of serendipity while Section III describes the demonstration plan.

II. System & Algorithms

A. Back-end Server

The back-end sever of our system implements the algorithms for selecting a set of recommended POIs. It extends the `MPG` server [5] and implements all the algorithms mentioned above: DisC Diversity [9], K-Medoid, a PageRank-based recommendation engine, and *PrefDiv*'s variations proposed in the `MPG` system [4] and in this paper.

Range Queries: One of the main operation in the algorithms implemented is to generate a nearest neighbor set. While several indexing schemes can be used, we utilize the *M-tree* spatial index [1], which is a balanced tree index designed to handle multi-dimensional dynamic data in general metric spaces, using the triangle inequality for efficient range queries.

Ranking: A set of intensity values for the venues to be recommended are defined based on different *objectives* [4]. For example, the distance-based intensity value I_d^v for a venue v can be obtained by considering the distance d between the current location q of the user and v. We can also define a popularity-based intensity value I_p^v by considering the number

1063-6927/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDCS.2017.286

of visitations to venue v, while additional popularity information can be considered using the Page Rank score π_v of venue v in a venue flow network. We also define a preference-based intensity value I_u^v based on a hierarchical user profile.

Combine above intensity values togethor, we have the venue ranking function $I_{p,d,u}^v$, which is a composite function that consists of three components, popularity-based intensity value, distance-based intensity value and preference-based intensity value:

$$I_{p,d,u}^v = \lambda * [(\sigma * I_p^v) + ((1-\sigma) * I_{d,q}^v] + (1-\lambda) * I_u^v \quad (1)$$

Diversity: The (dis)similarity between two venues is measured using two similarity distances: a *syntactic* distance based on the category structure of venues in Foursquare (the largest open venue database to date) and a *semantic* distance based on the venue name.

Category Tree: The *category tree* is built to capture the category structure of venues in Foursquare. Each internal node in the category tree represents a type of venue, which is a subcategory of the parent node with each leaf node representing the actual venue.

Word2Vec: Although the category tree is able to measure the similarity between two venues, this measurement is not very accurate as it cannot distinguish the difference between two venues that are under the same subcategory. In order to overcome this limitation, we utilize the Word2Vec framework [10] and store all the word vectors in memory as a hash map for efficient querying.

Combine both Category Tree and Word2Vec, we have the semantic distance function $Sim(v_i, v_j)$ for a given pair of venues v_i and v_j that measures the semantic distance between two given venues, which is essential for constructing a semantically diverse set of recommendations:

$$Sim(v_i, v_j) = \alpha * \underset{Tree}{Sim}(v_i, v_j) + (1-\alpha) * \underset{Vec}{Sim}(A, B) \quad (2)$$

where $\underset{Tree}{Sim}(v_i, v_j)$ is the semantic distance generated by the category tree, $\underset{Vec}{Sim}(A, B)$ is the semantic distance generated by the Word2Vec, A and B capture the vector representation of venues v_i and v_j respectively, and α is a tunable parameter that controls the weights between the category tree and Word2Vec.

Serendipity: To incorporate the serendipity with our recommendation algorithms, we employed two layers, namely, Serendipity of Venues and Serendipity of Routes.

Serendipity of Venues: To achieve serendipity of venues, we designed a probabilistic version of the *PrefDiv* algorithm [3], namely, *pPrefDiv* that introduces randomness in the selection of venues to incorporate the serendipity. Particularly, *pPrefDiv* differs from *PrefDiv* in the following fashion: when a venue x is qualified to be one of the recommendations for a range query q under *PrefDiv*, instead of including x into the result set, *pPrefDiv* decides whether x can be added to the result by using the combined intensity value of x as the way to determine its acceptance, such that the probability of a venue x being accepted is:

$$p(x \text{ is accepted}) = \frac{I(x)}{\text{Argmax}_{i:V}\, I(i)} \quad (3)$$

where $I(x)$ is the combined intensity value of I_d^v, I_p^v and I_u^v of a venue x, and V is the set of all venues within q. If x is accepted, it would be presented as one of the recommendation. Otherwise, x will be discarded and *pPrefDiv* would proceed to the next venue.

Such strategy allows *pPrefDiv* to incorporate the serendipity into the venue recommendations, while still preserving the high intensity value and semantic distance feature of the *PrefDiv* algorithms.

Serendipity of Routes: The algorithms described above provide a discrete set of venues that are both relevant to user's interests as well as semantically diverse with respect to each others. In order to support the recommendation of routes for k venues, where k is a user-defined variable, we utilize random walks to generate a set of initial routes.

The random walks are performed on a weighted graph between the venues. The weight w_e of edge e considers the distance between the current location of the user, the number of visits to the venue, and the user's preference towards a certain type of venue. Then the probability that our random walk will go through edge e is proportional to $1/w_e^\gamma$, where γ is a system parameter. The weight assigned is a tunable parameter that users can explore. Before generating the final route recommendations we perform a number of random walks (default value 50) to yield the initial candidate routes.

Using this initial set of routes we determine our final recommendations by introducing the concept of serendipity, which enables the platform to yield unexpected yet interesting results to the user. To measure the serendipity of a given route r, we first find the set of overlap venues O_{r,r_I^*} between r and the route that maximized the intensity value r_I^*. Based on the set of overlap venues we compute the *Overlap Factor (OvF)* between r and r_I^* as following:

$$OvF(r, r_I^*) = 1 - \frac{|O_{r,r_I^*}|}{|r|} \quad (4)$$

Then we compute the *Normalized Longest Common Subsequence* (*NLCS*) between r and the route that maximized the intensity value r_I^* as follows:

$$NLCS(r, r_I^*) = \frac{|L_{r,r_I^*}|}{|r|} \quad (5)$$

Thus, the serendipity σ_r of r would be:

$$\sigma_r = (1 - NLCS(r, r_I^*)) * OvF(r, r_I^*) \quad (6)$$

Pareto Front: Rather than optimizing over a single dimension our system first computes the Pareto Front of the random walks generated based on their serendipity σ_r, as described above and their diversity δ_r [4]. The Pareto Front includes the non-dominated points, i.e., a route r is part of the Front *iff*: $\sigma_r \geq \sigma_{r'}$ and $\delta_r \leq \delta_{r'}$, $\forall r' \in \mathcal{R}$, where $\mathcal{R}$ is the set of the random walks. Given that the Pareto Front might include a large number of routes, we divide it into n equal parts (through projections on the dimension with the maximum range) and choose a representative route for each part of the front based on the shortest distance to be covered (see Fig.1).

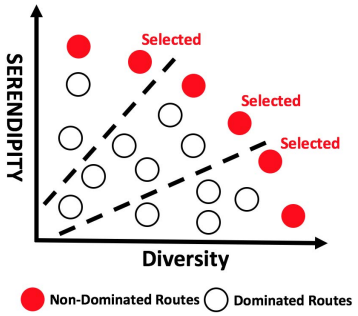

Fig. 1: Illustration of the Pareto Front.

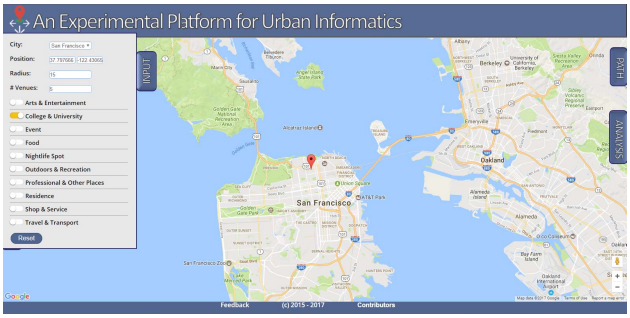

Fig. 2: Input Panel

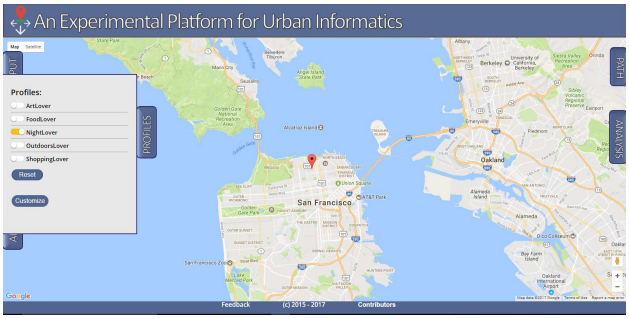

Fig. 3: Profile Panel

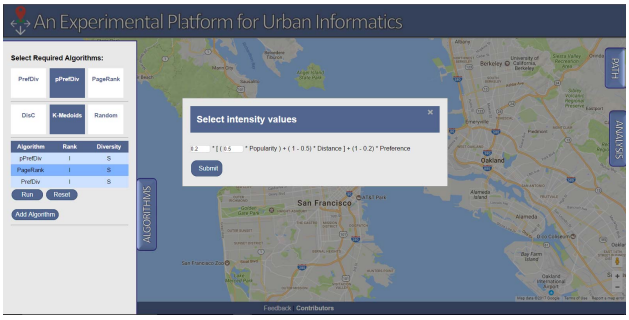

Fig. 4: Algorithms Panel

B. *Front-end Application*

The front-end is implemented using the Google Maps API for visualizing the results on a map. It currently supports the cities of New York and San Francisco. The recommended POIs are numbered and colored to match the number and the color of the algorithm making the recommendation, along with different provisions to drive or walk to those recommendations. The interface consists of five different panels: "Input", "Profile", "Algorithms", "Path" and "Analysis" (see Figs. 2 - 4).

The "Input" panel provides the options for the user to specify the inputs that describe the basic information for each query, such as the radial distance they are willing to travel, the number of venues they would like to get returned, and the types of venues they are interested in exploring (Fig. 2).

The "Profile" panel provides the options for the user to specify their own preferences by selecting any of the predefined profiles (i.e., *ArtLover, FoodLover, OutdoorsLover*, etc.) (Fig. 3), or to customize a selected preference profile by adjusting the values on the corresponding entry of the category tree.

The "Algorithms" panel (Fig. 4) allows users to choose or upload the recommendation algorithm(s) that would be involved in the location query. The algorithms are listed in the form of buttons. Each algorithm can be clicked and it then dynamically gets selected. Further, users can choose to design and upload their own venue recommendation algorithms by providing the name and the corresponding template java program. For each algorithm that is involved in the query, an option for adjusting the composition of the ranking and semantic distance function has also been provided. When the rank button (I button) is clicked for any selected algorithm, the venue ranking function (Eq. 1) will be presented to allow customization of the scaling parameters. By adjusting the value for scaling parameters λ and σ users/researchers are able to exploration the different composition of the three components. The values must between the range of 0.0 and 1.0 for both λ and σ. Similarly, the user/researcher would be presented with a formula of the semantic distance function (Eq. 2) when they click on the diversity button (S button), allowing them to adjust the scaling parameters α.

The "Analysis" panel visualizes the performance characteristics of the recommended venues from the selected algorithms in tabular form as well as in a scatterplot. The listed characteristics in terms of quality are the relevance score of the recommended venues, their diversity and the radius of gyration for each set of the recommended venues. We also report the run time taken for each algorithm as an indicator of interactivity. Further, this panel also has an option called "Compare" which enables the user to compare the results of the present query with the previous one.

The "Path" panel allow users to select the construction of the routes. For each set of recommended venues our prototype system is able to construct different routes based on different routing algorithms, such as random walk-based routing that incorporates the serendipity, distance optimized routing that seeks the minimum physical travel distance between each venue, as well as the relevance based routing that constructs the route based on the ranking of each recommended venues. Further, the "Path" panel allows users to build a customized trip based on a subset of recommended venues. To build a customized route the user would need to select the markers in the order that represents how they want to visit those venues. For example, (as shown in Fig. 6) the user selects markers one, four and five out of a total of five markers that were returned. After the selection is made, the segments of the customized route (i.e., user to marker one, marker one to

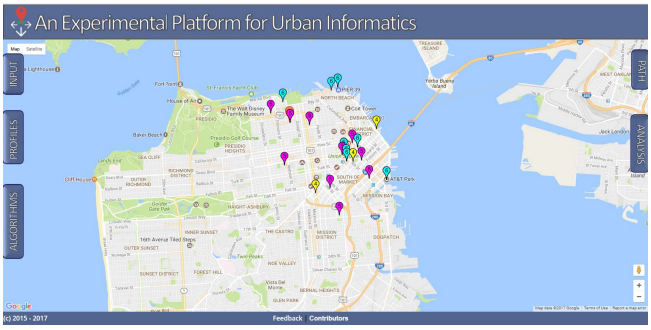

Fig. 5: Sample results of color-coded location markers based on the algorithm used for making the recommendation

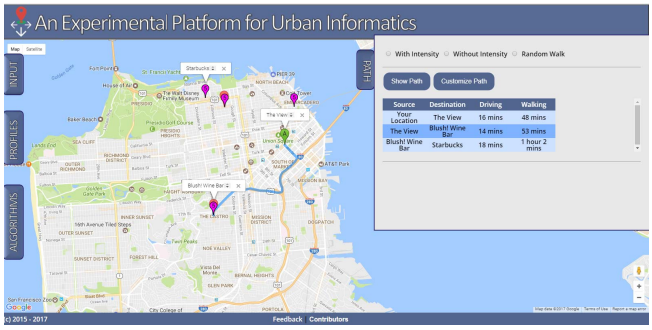

Fig. 6: Example Scenario

marker four and marker four to marker five) are shown in a table. These directions will also have the corresponding travel times displayed based on the type of the transportation that the user preferred (e.g., walking or driving). The support for both forms of transportation provides the user with more precise control over their journey.

III. Demonstration Plan

During the demonstration, we will run the front-end interface of the system on one or more laptops, while the backend will be hosted on a remote server. The participants will be given the opportunity to interact with the application not only as an end-user but also as an experimental researcher. We will demonstrate exactly how the user will be able to play each of these roles.

Application end-user: In this scenario, attendees will experience the effectiveness of our system through the view of an ordinary user. Specifically, the user will be able to initiate a location range query by entering their desired coordinates (longitude and latitude) in the respective fields or by dragging the location marker to the specified location on the map. Attendees can use the "Input" panel to provide additional information for their query (Fig. 2). Once the query has been defined attendees can then use the "Profile" panel to specify their preference for the query (Fig. 3). Finally, attendees can choose one or more algorithms among the different ones currently implemented. Once the algorithms for the experiment are chosen, they can submit the POIs recommendation query for execution by clicking the Run button in the "Algorithms" panel. The POIs returned as results will be visualized on the map with color-coded location markers based on the algorithm used for making the recommendation (See Fig. 5). After the results are plotted, through the "Path" panel, the end user would have the options to construct tours (i.e., routes) based on the venues suggested by the venue recommendation algorithms that are involved in the location query. Also, through the "Path" panel, users can get a recommendation of a serendipity route based on our random walk scheme.

Experimental researcher: In this scenario, we will demonstrate the ability of the platform to explore and compare the trade-offs between different parameter configurations and recommendation algorithms. This would enable more advanced users (i.e., researchers) to explore the characteristic of different algorithms and parameters. In particular and in addition to the functions described above for the end-users, our prototype system provides a higher level of customizability that allows researchers to explore different venue recommendation approaches. For instance, after the researcher selects a set of algorithms that are involved in the location range query and these algorithms would be displayed in a table beneath the algorithm selection menu (as shown in Fig. 4). This table contains the following information for each algorithm: Algorithm Name, a Rank button (shown as I after the algorithm name), and a Diversity button (shown as S next to I after the algorithm name). By clicking the corresponding rank or diversity button of an algorithm, the researcher can adjust the composition of the ranking (Eq. 1) and semantic distance (Eq. 2) function by changing the parameter that defines the weights of each component of both functions for that specific algorithm. These tunable parameters provide users with the ability to explore and perform sensitivity analysis over different relevance and diversity configurations. Furthermore, researchers can take advantages of advanced features mentions in Section II, such as the venue sub-categories, result dashboard and scatterplots that enables a higher level of customization and exploration.

References

[1] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, pages 426–435, 1997.

[2] E. Galbrun, K. Pelechrinis, and E. Terzi. Urban navigation beyond shortest route: The case of safe paths. *Information Systems*, 57:160–171, 2016.

[3] X. Ge, P. K. Chrysanthis, and A. Labrinidis. Preferential diversity. In *ExploreDB*, pages 9–14, 2015.

[4] X. Ge, P. K. Chrysanthis, and K. Pelechrinis. Mpg: Not so random exploration of a city. In *IEEE MDM*, 2016.

[5] X. Ge, S. R. Panati, K. Pelechrinis, P. K. Chrysanthis, and M. A. Sharaf. In search for relevant, diverse and crowd-screen points of interests. In *EDBT*, 2017.

[6] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi. Customized tour recommendations in urban areas. In *ACM WSDM*, 2014.

[7] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In *CIKM*, 2010.

[8] D. Levinson. A random walk down main street. In *Journal of Land Use, Mobility and Environment*, number 9, pages 31–40, 2016.

[9] E. P. Marina Drosou. Multiple radii disc diversity: Result diversification based on dissimilarity and coverage. *ACM TODS*, 40(1), 2015.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[11] D. Quercia, R. Schifanella, and L. M. Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *the 25th ACM conference on Hypertext and social media*, pages 116–125. ACM, 2014.

[12] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing popular routes from uncertain trajectories. In *ACM KDD*, 2012.