EPUI: Experimental Platform for Urban Informatics

Xiaoyu Ge[1], Panos K. Chrysanthis[1], Konstantinos Pelechrinis[2] and Demetrios Zeinalipour-Yazti[3]
[1] Department of Computer Science, [2] Department of Informatics & Network Systems, University of Pittsburgh, USA
[3] Department of Computer Science, University of Cyprus, Cyprus
[1] {xiaoyu, panos}@cs.pitt.edu, [2] kpele@pitt.edu, [3] dzeina@cs.ucy.ac.cy

ABSTRACT

Recent studies in urban navigation have revealed new demands (e.g., diversity, safety, happiness, serendipity) for the navigation services that are critical to providing useful recommendations to travelers. This exposes the need to design next-generation navigation services that accommodate these newly emerging aspects. In this paper, we present a prototype system, namely, EPUI (an Experimental Platform of Urban Informatics), which provides a testbed for exploring and evaluating venues and route recommendation solutions that balance between different objectives (i.e., demands) including the newly discovered ones. In addition, EPUI incorporates a modularized design, enabling researchers to upload their own algorithms and compare them to well-known algorithms using different performance metrics. Its user interface makes it easily usable by both end-user and experienced researchers.

ACM Reference Format:
Xiaoyu Ge[1], Panos K. Chrysanthis[1], Konstantinos Pelechrinis[2] and Demetrios Zeinalipour-Yazti[3]. 2018. EPUI: Experimental Platform for Urban Informatics. In *SIGMOD'18: 2018 International Conference on Management of Data, June 10–15, 2018, Houston, TX, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3183713.3193560

1 INTRODUCTION

As mobile computing and GPS technology increasingly becomes a part of our daily activities, various Points-of-Interest (POIs) or venue recommendation and route-planning services have been proposed to assist people effectively navigating through a city. Existing city navigation systems often focus on optimizing well-known efficiency objectives, such as minimizing the distance covered, maximizing the benefit obtained from the route as captured by some measure of venue quality, and so on (e.g., [11, 12, 17]). As recent works (e.g., [3, 7, 9, 14]) aim to better understand what constitutes a true user-centric recommendation and develop novel recommendation algorithms that better align with the individual user's interests, it is critical to have an efficient tool that helps to discover, compare, converge, and evaluate different aspects and approaches of urban navigation.

Towards this end, we developed an experimental platform, called *EPUI (Experimental Platform of Urban Informatics)*, which provides a fully customizable framework for urban informatics experimentations. It currently supports the study of different urban navigation approaches. EPUI incorporates many existing approaches for venues recommendation and route construction as well as optimizing structures such as spatial indexing and weighted route network graphs, which are adjustable by the user. EPUI allows users to upload their own recommendation algorithms, indexing structures and evaluation metrics. Its user-friendly front-end interface enables a user to compare the result of different recommendation approaches by both presenting the recommended venues and routes visually on a map as well as displaying evaluation metrics through summary dashboards.

In addition to being a research tool, EPUI can also be used as a demonstration platform, capable of providing informative venue and route recommendations to the audience. In particular, for venue recommendation, EPUI consider three different aspects *relevance*, *diversity* and *serendipity* and implements a set of existing recommendation algorithms (e.g., DisC Diversity [6], K-Medoid, PrefDiv [8] and PageRank). For route constructions, EPUI employs an advanced route construction algorithm, optimizing multiple simultaneous objectives (e.g., distance, safety, happiness) [7].

2 SYSTEM & ALGORITHMS

2.1 Back-end Server

The back-end server of EPUI is implemented in Java with a modularized design, which allows the user to customize its components, including the algorithm and the optimization objectives of both venue and route recommendation, as well as indexing structures used for range queries. Below we briefly discuss each of EPUI's main components.

Range Queries: One of the main operation in the algorithms implemented is to generate the nearest neighbor set. While several indexing schemes can be used, the default is the *M-tree* spatial index [5]. M-tree is a balanced tree index that is designed to handle a large scope of multi-dimensional dynamic data in general metric spaces and it uses the triangle inequality for efficient range queries similar to those required in EPUI.

Ranking: A set of intensity values for the venues to be recommended are defined based on different *objectives* [9]. For example, the distance-based intensity 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 define a popularity-based intensity I_p^v by considering the number 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 I_u^v based on a hierarchical user profile.

By combining the above intensity values together, we generate the venue ranking function $I_{p,d,u}^v$, which is a composite function

SIGMOD'18, June 10–15, 2018, Houston, TX, USA

ACM ISBN 978-1-4503-4703-7/18/06…$15.00
https://doi.org/10.1145/3183713.3193560

of popularity-based intensity, the distance-based intensity and the preference-based intensity:

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

Diversity: Diversity as a measure of (dis)similarity between two venues is measured using two similarity distances: a *syntactic* distance based on a category structure of venues and a *semantic* distance based on the venue name.

Category Tree: 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. The category tree is currently built to capture the category structure of venues in Foursquare (the largest open venue database to date), but it can evolve to include new venues from other sources as well.

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 [13] 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: We incorporated serendipity within our recommendation algorithms thru some form of randomization.

Serendipity of Venues: To achieve serendipity of venues, we designed a probabilistic version of the *PrefDiv* algorithm [8], 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^{v}_{d}, I^{v}_{p} and I^{v}_{u} 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* algorithm.

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 other. In order to support the recommendation of routes for k venues, where k is a user-defined variable, we utilize random walks [10] to generate a set of initial routes *SR*. To construct a useful route recombination with serendipity each candidate routes of *SR* will be evaluated with respect to user's preferred optimization criteria.

Multi-objective Routing To support construction of routes that go beyond just the shortest distance, our EPUI employs a route network of each supported city and models it as a weighted directed graph $G = (V, E)$. The nodes of the graph V, represent intersections, while the edges E, represent the road segments that connect intersections. The direction of the road is used to determine the accessibility of the transportation mode that a user utilizes (e.g., car, bicycle, walking). We extract such road network from a city using a crowdsourcing-based open platform call OpenStreetMap [1].

When providing route recommendations, EPUI first relies on the specified routing methods (e.g., HighestRelevance, Random Walk) to determine the sequence that each venue should be visited. Based on the visiting sequence of each venue, EPUI utilizes G to produce the actual route that concatenates each venue. In particular, EPUI relies on the weighted edges $e \in E$ to determine the actual path that the user would take when traveling from one venue to the next. In our route network graph G, each road segment $e \in E$ can be associated with up to two (unrelated) types of user-defined weights. This allows advanced users (e.g., researchers) to exploit the route network graph module of EPUI to construct route recommendations that are optimized towards one or more aspects of the city (e.g., length, safety, happiness). To enable such user-defined weights, EPUI utilizes a modularized design that facilitates the assignment of the user-defined weights to each edge e of G. Consequently, to produce routes that are optimized towards the user's desired optimization criteria, users can specify function $f(e)$ that returns a weight $w \in R^{+}$ for each edge e. Note that currently up to two criteria may be specified simultaneously, and in such case a bi-criteria optimization problem would be formulated. However, this can be easily extended to multiple ones to include for example safety, happiness and trajectory preference, either derived from user behavior [4, 18], or explicitly specified [15, 16].

Serendipity without threats As an illustration, we integrated our serendipity-based navigation with the Safe Path navigation [7]. More specifically, EPUI provides *length*, denoted by $\iota(e)$, and *risk*, denoted by $r(e)$, as two examples of the user-defined weights to be assigned to each edge of G. The length of e simply represents the actual distance between the two intersections connected by the associated road segment, and the risk can be interpreted as the probability of a crime being committed on that segment.

We further adopted Safe Path's risk model for the urban road network that assigns a risk score r to each edge that is proportional to the probability of a crime happening on the corresponding street segment. To assign the risk score r on each route segment e we make use of open crime data made available by the city governments (e.g., [2]). To obtain the risk scores/model we first extract a sample set of discrete latitude/longitude points from the route segment denoted as $\Gamma(e)$, which essentially provides information about the actual geographic shape of the street segment. Then, we use the geographic coordinates of the crime incidents to compute

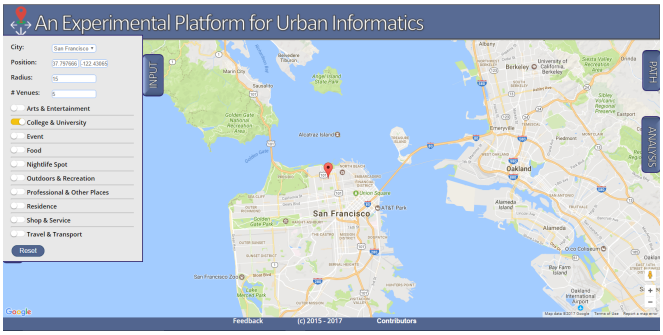

Figure 1: Input Panel

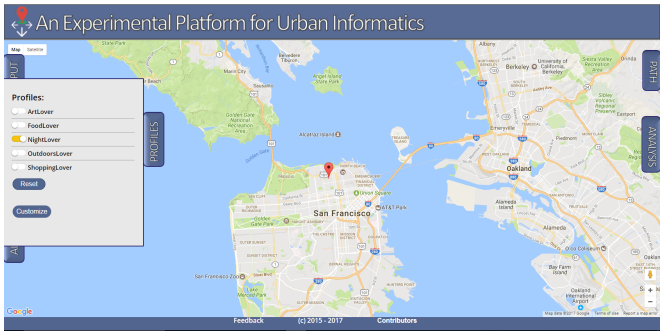

Figure 2: Profile Panel

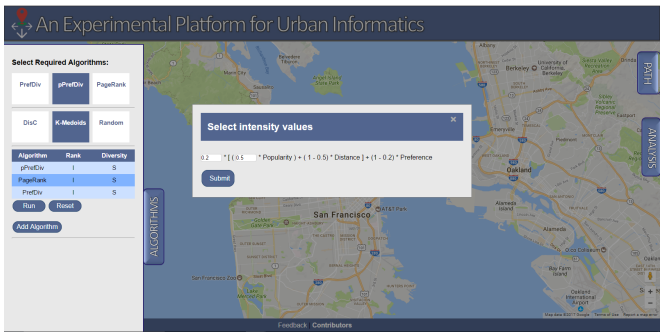

Figure 3: Algorithms Panel

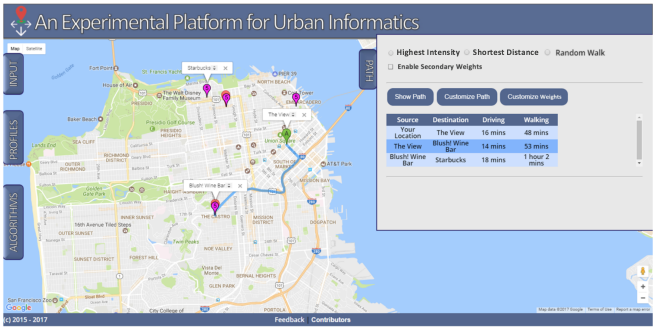

Figure 4: Display Results

a spatial density for the criminal activity by applying a Gaussian kernel density estimation (KDE). Given n points of crime incidents $c_1, c_2, ..., c_n$ on a 2-dimensional plane, the Gaussian kernel estimates the density of criminal activity at point p as:

$$\lambda(p) = \frac{1}{nh^2} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp(-\frac{\|c_i - p\|_2^2}{2h^2}) \quad (4)$$

where $\|c_i - p\|_2^2$ is the Euclidean distance between points c_i and p and h is the bandwidth used. The bandwidth h dictates the spread of the Gaussian kernel that is centered at each data point and hence, it controls the smoothness of the estimated density.

Once the density of criminal activity of one point is estimated, we evaluate it on the actual road segments. For this purpose, we make use of the geometry $\Gamma(e)$ of each edge e of the road network. Evaluating Eq. (1) on every point $\xi \in \Gamma(e)$ and summing up we obtain the crime activity density $\Lambda(e)$ on road segment e, such that $\Lambda(e) = \sum_{\xi_i \in \Gamma(e)} \lambda(\xi_i)$. Here, $\Lambda(e)$ is proportional to the probability of observing a crime incident on edge e. Thus, we compute the risk weights for every edge e is computer as the normalized densities:

$$r(e) = \frac{\Lambda(e)}{\sum_{e' \in G} \Lambda(e')} \quad (5)$$

Having both length and risk as two independent weights on G allows the formulation of a bi-criteria optimization problem, where the goal is to minimize both the length and the risk of the reported path. To effectively solve such bi-criteria optimization problem, EPUI utilizes the state-of-the-art algorithms from [7] to determine the Pareto Front for the risk vs. distance trade-off and obtain a small set of non-dominated paths that summarize the (potentially infinite) solution space and provide different levels of trade-offs between the two objectives as the final recommendation. By enabling users to specify their own user-defined weights, EPUI allows users to formulate both single-objective and bi-criteria optimization problem with various aspects of the city.

2.2 Front-end Application

The front-end is constructed from javascript and PHP with the help of Google Maps API for visualizing the results on a map. It communicates with the back-end server through JSON, and currently supports the cities of New York, San Francisco and Pittsburgh. The recommended venues or 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" (Figs. 1-3).

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. 1).

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. 2), or to customize a selected preference profile by adjusting the values on the corresponding entry of the category tree.

The "Algorithms" panel (Fig. 3) allows users to choose, customize and upload the recommendation algorithm(s) that would be involved in the location query. To upload an algorithm, EPUI simply

requires the user to providing the name and the corresponding template java program, and then it will be include in the algorithms list along with other existing algorithms. For each algorithm, the user has an option for adjusting the composition of the ranking (Eq. 1) and semantic distance (Eq. 2) function.

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.

The "Path" panel (Fig. 4) allows users to select the construction method of the routes based on sets of recommended venues and a route network graph G. Once a route construction method has been selected, it would determine the visit sequence of each recommended venues. Later the route that connects each venue (according to the specified sequence) would be constructed based on the weights assigned on G. Furthermore, users is able to assign any weights to the edges of G by upload their customized weight modules as well as specify their desired trade-off between weights that are currently assigned to each edge of G. Finally, this panel also includes a statistics table that would display basic informations (e.g., originate, destination) or evaluation measures (e.g., distance, risk, relevance, diversity) according to user's configuration.

3 DEMONSTRATION PLAN

During the demonstration, we run the front-end interface of the system on one or more laptops, while the backend is hosted on a remote server. The participants are given the opportunity to interact with the application as an experimental researcher. We demonstrate exactly how the user is able to play this role.

During the demonstration, attendees are 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. 1). Once the query has been defined attendees can then use the "Profile" panel to specify their preference for the query (Fig. 2). In the "Algorithm" panel 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 are being visualized on the map with color-coded location markers based on the algorithm used for making the recommendation (See Fig. 4). 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 recommended venues with various of objectives (e.g., distance, relevance, serendipity, safety).

Furthermore, attendees can experience EPUI's ability to explore and compare the trade-offs between different parameter configurations and recommendation algorithms. As mentioned previously in Sec. 2, EPUI provides a higher level of customizability that allows researchers to explore different venue recommendation and route construction approaches. For instance, after attendees select a set of algorithms that would be involved in the query, these selected algorithms would be displayed in a table beneath the algorithm selection menu. (as shown in Fig. 3). For each algorithm, attendees can adjust the composition of the ranking (Eq. 1) and the semantic distance (Eq. 2) by changing the parameter that defines the weights of each component of both functions for that specific algorithm. These tunable parameters provide attendees with the ability to explore and perform sensitivity analysis over different relevance and diversity configurations. Attendees can also experience the benefit of EPUI's modularized design through the integration of new venues recommendation algorithms that are not originally supported by EPUI.

In addition, researchers can take advantages of advanced features mentions in Sec. 2, such as the venue sub-categories, result dashboard and scatterplots that enables further customization and exploration. Once the set of recommended venues have been obtained, attendees would have the option to customize the route construction method to obtain routes that suit their preferences. Specifically, they would be able to select one or more aspects (e.g., distance, relevance, diversity, serendipity, and safety) as the optimizing criteria to produce routes that are optimized towards these selected objectives.

Acknowledgment: We would like to thank Ameya Daphalapurkar, Manali Shimpi, Chihao Sun and Darpun Kohli for their help in the development of EPUI as part of their MS and First Experience in Research program, as well as the anonymous reviewers. This work was partially supported by NSF CPS-1739413 and NIH U01HL137159 Awards. This paper does not represent the views of NSF and NIH.

REFERENCES

[1] Openstreetmap. http://www.openstreetmap.org, 2017.
[2] Pittsburgh police arrest data. https://catalog.data.gov/dataset/pittsburgh-police-arrest-data, 2017.
[3] V. Ceikute and C. S. Jensen. Vehicle routing with user-generated trajectory data. In *16th IEEE MDM*, 2015.
[4] Z. Chen, H. T. Shen, and X. Zhou. Discovering popular routes from trajectories. In *IEEE ICDE*, 2011.
[5] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, 1997.
[6] M. Drosou and E. Pitoura. Multiple radii disc diversity: Result diversification based on dissimilarity and coverage. *ACM TODS*, 40(1), Article No. 4, 2015.
[7] E. Galbrun, K. Pelechrinis, and E. Terzi. Urban navigation beyond shortest route: The case of safe paths. *Information Systems*, 57(C):160-171, 2016.
[8] X. Ge, P. K. Chrysanthis, and A. Labrinidis. Preferential diversity. In *ACM ExploreDB*, 2015.
[9] X. Ge, P. K. Chrysanthis, and K. Pelechrinis. Mpg: Not so random exploration of a city. In *IEEE MDM*, 2016.
[10] X. Ge, A. Daphalapurkar, M. Shimpi, D. Kohli, K. Pelechrinis, P. K. Chrysanthis, and D. Zeinalipour-Yazti. Data-driven serendipity navigation in urban places. In *IEEE ICDCS*, 2017.
[11] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi. Customized tour recommendations in urban areas. In *ACM WSDM*, 2014.
[12] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In *ACM CIKM*, 2010.
[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.
[14] D. Quercia, R. Schifanella, and L. M. Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *ACM HT*, 2014.
[15] S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng, and P. Kalnis. User oriented trajectory search for trip recommendation. In *EDBT*, 2012.
[16] S. Shang, R. Ding, K. Zheng, C. S. Jensen, P. Kalnis, and X. Zhou. Personalized trajectory matching in spatial networks. *The VLDB Journal*, 23(3):449-468, 2014.
[17] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing popular routes from uncertain trajectories. In *ACM KDD*, 2012.
[18] J. Yuan, Y. Zheng, X. Xie, and G. Sun. T-drive: Enhancing driving directions with taxi drivers' intelligence. *IEEE TKDE*, 25(1):220–232, 2013.