Semantically Diverse Paths with Range and Origin Constraints

Xu Teng
Iowa State University
Ames, Iowa, USA
xuteng@iastate.edu

Goce Trajcevski
Iowa State University
Ames, Iowa, USA
gocet25@iastate.edu

Andreas Züfle
George Mason University
Fairfax, Virginia, USA
azufle@gmu.edu

ABSTRACT

One of the most popular applications of Location Based Services (LBS) is recommending a Point of Interest (POI) based on user's preferences and geo-locations. However, the existing approaches have not tackled the problem of jointly determining: (a) a sequence of POIs that can be traversed within certain budget (i.e., limit on distance) *and* simultaneously provide a high-enough diversity; and (b) recommend the best origin (i.e., the hotel) for a given user, so that the desired route of POIs can be traversed within the specified constraints. In this work, we take a first step towards identifying this new problem and formalizing it as a novel type of a query. Subsequently, we present naïve solutions and experimental observations over a real-life datasets, illustrating the trade-offs in terms of (dis)associating the initial location from the rest of the POIs.

CCS CONCEPTS

• **Information systems → Geographic information systems**.

KEYWORDS

POI sequence, Range-constraint, Diversity, Origin

ACM Reference Format:
Xu Teng, Goce Trajcevski, and Andreas Züfle. 2021. Semantically Diverse Paths with Range and Origin Constraints. In *29th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '21), November 2–5, 2021, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3474717.3483985

1 INTRODUCTION

The advances in networking and location-aware devices [11] have enabled the generation of large volumes of data providing unique opportunities for coupling geo-spatial data (e.g., locations, distances) with additional context such as properties of venues, users preferences, popularity, etc. Contrary to the the traditional recommendation systems capable of suggesting items like books, movies, and other products – the Location-Based Recommendation Systems (LBRS) [1] fuse them with the functionalities of efficient location-aware spatial queries processing commonly available by the Location-Based Services (LBS) [8].

One paradigm that has gained popularity in the past decade are the, so called, *semantic* (synonymously, *activity* or *spatio-textual*) trajectories, which interleave mobility data with the properties of activities (e.g., transportation mode, stay) and the Points of Interest (POIs) [12]. In addition to their own challenges in terms of spatio-textual queries processing [5], combining sequences of visits, activities and properties of POIs with data obtained from Location-Based Social Networks (LBSN) has enabled further improvements on the quality of recommendations [2]. However, works targeting the recommendation of a set of POIs to users [19] do not provide routes that maximize the number and/or variety of POIs that can be visited within a given travel-budget. A particular facet of the problem of recommending a set of POIs (as well as a path (or, a sequence) of POIs [18]) is to incorporate the (semantic) *diversity* – in the sense that the user is enabled to experience POIs with greater variety of their combined features [16]. Additional types of constraints that have been considered include the limit (in terms of travel-time or travel-distance) that the user may have, along with the number of POIs in the "budget" [15].

At the heart of the motivation for this work is the observation illustrated by the following:

Example 1. *Alice is attending a conference finishing on November 5th, and her return flight is early on November 7th. As a conscientious attendee, she will not have any time to explore any attractions in the venue-city during the conference. She wants to use November 6th for that purpose. Having only a single day, instead of directly selecting (locations of) specific POIs, she would like to simply put a limit on the trip for visiting them (as she needs to have a business diner meeting with folks whom she met at the conference), and provide a few categories of POIs that she would be interested in visiting. Her additional objective is to select a hotel from which there exists a path that allows her to visit such POIs within the allocated time.*

It may be tempting to exploit the existing results on k-Nearest Neighbor (kNN) for trajectories with respect to static points (e.g., [7]), or recent works on query processing in activity/semantic trajectories [14] to help Alice. However, we note that the nature of the problem illustrated in Example 1 is rather different: unlike the spatial and spatio-temporal variants, Alice does not specify in advance the precise sequence of points to be visited. In addition, the existing approaches which incorporate semantic aspect in the motion, do so only for known trajectories and do not cater to the diversity aspect. Complementary to these, the works incorporating the diversity of the POIs semantic descriptors in motion planning (cf. [15]) have not addressed the issue of selecting the location before the trip takes place.

To our knowledge, this is a novel kind of query which combines Semantic Diverse Paths with Range and Origin Constraints ($\mathbf{SP}_O^R$). In addition to the scenario discussed in Example 1 and other tourist based scenarios, $\mathbf{SP}_O^R$ query is important in domains such as exploratory process control [13] where one would like to reason about different executional constraints (i.e., coupling duration with

SIGSPATIAL '21, November 2–5, 2021, Beijing, China

ACM ISBN 978-1-4503-8664-7/21/11.
https://doi.org/10.1145/3474717.3483985

diverse classes of events) by also suggesting a proper origin-state for a given analysis/simulation [3].

The main contributions of this work can be summarized as follows:

- We introduce and formally define a novel type of query – $\mathbf{SP}^R_O$, which can be used to determine path for a semantically diverse POI sequence *and* its origin, under semantics and range constraints.
- We provide real dataset which we use to demonstrate that naïve approaches for graph exploration that one may thing of using to answer $\mathbf{SP}^R_O$, cannot guarantee practically acceptable behavior.

2 PRELIMINARIES AND PROBLEM DEFINITION

We now formalize the problem statement.

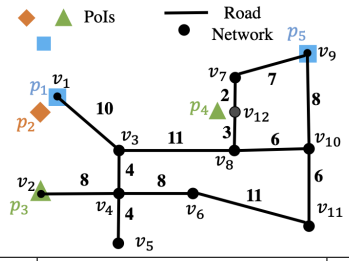
We consider a *POI-Augmented Road Network* as a weighted directed graph $\mathcal{G} = (V, E)$, where V is a set of vertices and each vertex $v \in V$ is associated with a (possibly empty) set of POIs $v.\mathrm{P}$; $E \subseteq V \times V$ represents the set of edges between pairs of vertices and each edge $e \in E$ is associated with a *weight* $e.\mathrm{W} \in \mathbb{R}^+$ representing the cost of traversing e.

In practice, $\mathcal{G}$ might be constructed from a regular road network graph $G = (V_{road}, E_{road})$ where edges in E_{road} correspond to road segments and the vertices V_{road} are their respective end-points, and each $v \in V_{road}$ is associated with a geo-location $v.\mathrm{L}$ such as (latitude, longitude) or (x, y) in a suitable coordinate system. Besides, we consider a collection of POIs $\mathcal{P} = \{p_1, ..., p_{|\mathcal{P}|}\}$, where each $p \in \mathcal{P}$ is characterized by geo-location $p.\mathrm{L}$, *descriptors* $p.\mathrm{D}$ and *category* $p.\mathrm{C}$.

To generate $\mathcal{G} = (V, E)$, we consider $G = (V_{road}, E_{road})$ and $\mathcal{P}$ as inputs and start with adding each $v \in V_{road}$ to V(initially $v.\mathrm{P} = \emptyset$), and each $e \in E_{road}$ to E. If the location of a particular POI $p \in \mathcal{P}$ coincides with a given $v \in V$ (i.e., $p.\mathrm{L} = v.\mathrm{L}$), we add p to $v.\mathrm{P}$. However, in practice, a particular POI might not be located at any $v \in V$, especially when POI dataset is obtained from a different source. In such a scenario, we employ map-matching strategy (cf. [4, 15]) to project a POI $p \in \mathcal{P}$ to the nearest location on the nearest edge in G (which, sometimes may coincide with a vertex in V). This projected location becomes a new vertex v_p and is added to V, initialized with $v_p.\mathrm{P} = \{p\}$. Additionally, a new vertex splits the associated edge $e = (v_i, v_j)$ into two new edges $e_i = (v_i, v_p)$ and $e_j = (v_p, e_j)$ added to E, e is removed from E.

Fig. 1 presents a small-scale example illustrating a POI Road Network with 12 vertices, 11 of which $(v_1, \ldots, v_{11})$ are part of the original road network. Five POIs $\mathcal{P} = \{p_1, p_2, p_3, p_4, p_5\}$ (shown as colored rectangles, triangles and diamond, representing three different categories) are augmented in the network. The locations of three of them (p_1, p_3 and p_5) coincide with locations of vertices from the road network. When map-matching p_2, it is assigned to v_1 and hence $v_1.\mathrm{P} = \{p_1, p_2\}$. However, p_4 is projected to a location v_{12} among the edge (v_7, v_8) and, consequently: (1). v_{12} is added to V with $v_{12}.\mathrm{P} = \{p_4\}$; (2). the edge (v_7, v_8) is split into (v_7, v_{12}) and (v_{12}, v_8), added to E; (3). the edge (v_7, v_8) is removed from E.

We reiterate that each POI is described by descriptors and a category (cf. the table in Fig. 1). Textual descriptors, such as restaurant menus, attraction reviews and keywords, are provided and give a sufficiently accurate description of corresponding POI. To extract latent topics from textual descriptors we leverage *Latent Dirichlet allocation* (LDA) based diversity [16] in this work to determine the latent category of each POI.



Attractions	Descriptors	Category
p_1	{Historical, Cultural, Art}	Museum
p_2	{Turtle, Tiger, Safari}	Zoo
p_3	{Fountain, Forest, Playground}	Park
p_4	{Green, Fish, Monument}	Park
p_5	{Literature, Art, Painting}	Museum

Figure 1: Example of POI Road Network

For a given POI road network $\mathcal{G} = (V, E)$, a path $\pi = [v_1, ..., v_{|\pi|}]$ is a sequence of adjacent vertices in $\mathcal{G}$, i.e., $\forall\ i = 1, ..., |\pi| : v_i \in V$ and $\forall\ i = 1, ..., |\pi| - 1 : (v_i, v_{i+1}) \in E$. For a given path π, its *cost* is defined as the sum of the weights of traversed edges $\pi.\mathrm{cost} := \sum_{i=1}^{|\pi|-1} e(v_i, v_{i+1}).\mathrm{W}$, and the set of POIs among a path $\pi.\mathrm{P} := \bigcup_{i=1}^{|\pi|} v_i.\mathrm{P}$ – i.e., the union of all the POIs contained in the vertices along π.

We note that a semantic path needs not be *simple*, i.e., it can have cycles and visit the same vertex more than once. This is necessary in scenarios where a path needs to collect a POI(s) which has only one incident edge (i.e., one adjacent vertex). In Fig. 1, $\pi = [v_8, v_{12}, v_8, v_3, v_1]$ is an example of a semantic path having cost $\pi.\mathrm{cost} = 3 + 3 + 11 + 10 = 27$ that includes the set of POIs $\pi.\mathrm{P} = \{p_4, p_1, p_2\}$.

Next, we present a few formal definitions.

Definition 1 (Categorical Diversity). *Let $\boldsymbol{C}$ denote a vector containing $|\boldsymbol{C}|$ components for a distinct category. cDiv is a diversity function mapping a given set of POIs P to a vector of length $|\boldsymbol{C}|$ where the value of each component corresponds to the number of its appearances in each category among all the POIs in P.*

Take the instance from POI road network in Fig. 1, where $\boldsymbol{C} = \langle \text{Museum, Park, Zoo} \rangle$. Given $\pi = [v_2, v_4, v_3, v_8, v_7, v_9]$, $cDiv(\pi.\mathrm{P}) = cDiv(\{p_3, p_4, p_5\}) = \langle 1, 2, 0 \rangle$ representing 1 museum and 2 parks.

Definition 2 (Semantically Diverse and Range Constrained Path). *Let $\mathcal{G} = (V, E)$ be a POI road network and $\boldsymbol{C}$ denote all categories among the POIs in $\mathcal{G}$. Given a positive value $\varepsilon \in \mathbb{R}^+$ and a vector $\overline{\theta} = \langle \theta_1, ..., \theta_{|\boldsymbol{C}|} \rangle$ where θ_i $(i = 1, ..., |\boldsymbol{C}|) \in \mathbb{N}$ represents the desired number of POIs in i^{th} category. A path π is called a* Semantically Diverse and Range Constrained Path *($\boldsymbol{SP}^R$) if:*

$$\pi.cost \leq \varepsilon$$
$$cDiv(\pi.P)_i \geq \theta_i \ (\forall\ i = 1, ..., |\boldsymbol{C}|)$$

The concepts introduced so far are shown in Fig. 1. Assume firstly that we merely focus on the distance range and consider a distance limit $\varepsilon = 25$. Possible π's could include (but not limited to) $\pi_1 = [v_4, v_2, v_4]$, $\pi_2 = [v_{10}, v_9, v_7, v_{12}]$ and $\pi_3 = [v_{10}, v_8, v_{12}]$. However, if $\overline{\theta}$ is configured to $\langle 1, 1, 0\rangle$ corresponding to $\boldsymbol{C} = \langle$Museum, Park, Zoo$\rangle$, that is to say we prefer at least one museum and one park, then π_2 is the only one satisfying $\mathbf{SP}^R$.

We recall that in addition to having semantically and "executionally" constrained paths, one may want to impose further constraints in the *origin*. For example, in addition to catering to tourist's varied interest in terms of POIs, one may want to recommend hotels to book.

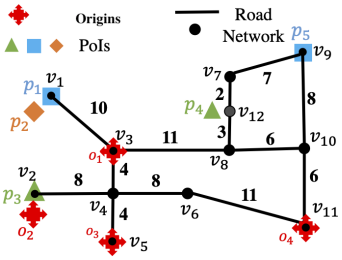


Figure 2: Example of POI Network with Origin locations

Towards that, we consider a subset O of the vertices in network (indicated with red crosses in Fig. 2) to be representatives of another restriction – possible *origins* for the path. Specifically, we are interested only in those $\mathbf{SP}^R$s whose starting location is an element of O as valid results. We call this novel (extended) variant $\mathbf{SP}^R$ – $\mathbf{SP}^R$with origin constraint. The assumption that a particular $o \in O$ is a vertex from POI road network comes without any loss of generality, as we can project any location to a (potentially new) network node (for instance, o_2 can be attached to v_2 in Fig. 2).

Definition 3 ($\mathbf{SP}^R_O$). *Let $\mathcal{G} = (V, E)$ be a POI road network, let $O \subseteq V$ be a set of candidate origin locations, let C be a set of POI categories, let $\varepsilon \in \mathbb{R}^+$ be a distance threshold, and let $\overline{\theta} \in \mathbb{N}^{|C|}$ be a vector of numbers of user desired POIs in each category. A Semantically Diverse Path with Range and Origin constraints ($\mathbf{SP}^R_O$) Query returns a $\mathbf{SP}^R$ from $o \in Q$.*

Recent results in [15] have provided a solution to discover a simplified version of $\mathbf{SP}^R$ in POI network, whereby the user must fix a single unique origin. However, in practice, there might be numerous candidates for a origin – e.g., a large number of hotels that can be chosen as starting locations – and the user desires a recommendation. Thus, the $\mathbf{SP}^R_O$ not only determines an $\mathbf{SP}^R$ satisfying all constraints, but also a preferred origin from O.

3 EVALUATIONS AND OBSERVATIONS

Since there are no existing dataset which can be directly used for evaluating any approaches targeting $\mathbf{SP}^R_O$ -like queries, one of the tasks for this initial stage was to integrate real data from several sources. To construct the POI network, we relied on two main resources: road network from OpenStreetMap, and attractions/POIs along with related reviews from TripAdvisor. The POI network of New York City, New York, USA contains 56, 730 nodes, 142, 233 edges and 621 POIs. In terms of POIs, on average, each POI has 52.41 reviews and each reviews contains 29.22 words. Regarding the origin locations O, 576 accommodations (other than 621 POIs mentioned above) in New York City have been considered. The data used to generate this POI network, as well as the code to generate POI network can be found published at https://github.com/XTRunner/SPRO-query.git.

Typically, when a problem involves path planning, one would like to capitalize on existing shortest path algorithms [10]. However, from a complementary perspective, when designing a path intending to visit a sequence of vertices on a graph (subject to certain criteria) – there is the intuition brought about by the Traveling Salesman Problem (TSP). In our case, we have the following:

Lemma 1. *Processing $\mathbf{SP}^R_O$ query in NP-hard.*

Proof. We reduce the NP-complete Traveling Salesman Problem (TSP) to a special of a $\mathbf{SP}^R_O$ query having $|Q| = 1$ (having a single query location), having $\theta = \mathbf{1} = (1, ..., 1)$ (requiring to visit all classes of POIs exactly once), having $\forall p_i, p_j \in \mathcal{P} : p_i \neq p_j \implies p_i.C \neq p_j.C$ (no two POIs have the same class), and having $\forall v \in V^{\mathcal{P}} : |v.P| = 1$ (each vertex has exactly one POI). The traveling salesman problem (TSP) decides if, for a graph $G_{TSP} = (V_{TSP}, E_{TSP})$ and a starting note $q_{TSP} \in V_{TSP}$, there exists a round trip of distance not greater than ε_{TSP} that visits all vertices in V_{TSP} at least once. This can be reduced to a special case of a $\mathbf{SP}^R_O$ query by setting $\mathcal{G}^{\mathcal{P}} = G_{TSP}$, $Q = q_{TSP}$, and $\varepsilon = \varepsilon_{TSP}$. If and only if this $\mathbf{SP}^R_O$ query returns a path, then the decision of the corresponding TSP is true (and false otherwise). Since TSP is an NP-complete problem [9], the problem of answering this special case of a $\mathbf{SP}^R_O$ query is NP-hard, and the general problem of answering $\mathbf{SP}^R_O$queries is also NP-hard. □

In this preliminary stage, we resort to exploring the benefits of certain popular heuristics and conduct a comparison evaluation between them. Dijkstra algorithm [6] is one of the most popular shortest-path algorithms. However, it does not consider any preference on the vertices among the retrieved path. In our settings of $\mathbf{SP}^R_O$query, one natural extension of Dijkstra algorithm would be greedily collecting the next nearest POI, which is demanded in $\overline{\theta}$, from the current location. From the origins O, Dijkstra algorithm is employed to find the nearest POI p, which is then used as the new initial location and we search for the next nearest POI from p. The whole greedy process will be terminated until the distance budget is exploited and we call it *Greedy-Dijkstra* searching strategy. Aside of them, another option we utilize in this work is *Random Walk with Restart* (RWR) [17]. RWR starts from a random vertex among the origins O and creates a random path using adjacent edges until

(1) the generated path satisfies all POI categories in $\overline{\theta}$. In this case, the path is returned;
(2) the length of the generated path is greater than ε. In this case, the algorithm restarts;
(3) a time limit is reached. In this case, None is returned.

For our experiments, we use $|C| = 6$ different POI categories corresponding to the number of latent topics yielding the highest topic coherence. To thoroughly evaluate different searching algorithms,

# of Origins	Found?	Dijkstra	Greedy-Dijkstra	RWR ×3
576	Y	928	975	509
	N	134	87	553
200	Y	894	947	701
	N	168	115	361
60	Y	830	875	645
	N	232	187	417

Table 1: Comparison of results

177 different category preferences $\overline{\theta}$ (i.e., the number of POIs preferred which are distributed in different categories) were randomly generated by (1) choosing an integer k uniformly at random between 1 and 5 as the user desired number of POIs, and (2) choosing k categories with replacement from C. In terms of distance budget, 500, 700, 1000, 1500, 2500 and 3500 meters are evaluated. Moreover, we realize the pool of origins might as well influence the performance. Thus, we evaluate experiments having all 576 origins, as well as subsets of random samples of size 200 and 60.

The experiments are conducted on a PC with Intel(R) Xeon(R) CPU *E*3-1240 v6 @3.70GHz and 32GB RAM. Windows 10 Enterprise 64-bit is the operating system, and all the algorithms are implemented by Python 3.7. Both the POI network and code are available at https://github.com/XTRunner/SPRO-query.git.

Table. 1 illustrates the effectiveness benefits (and, implicitly, the efficiency trade-off) between three most popular searching strategies introduced above. Specifically, the first column shows the number of origins. The values of the second column are labels 'Y' or 'N', indicating whether any path satisfied $\mathbf{SP}_O^R$query was Found or Not, respectively. Finally, for each row corresponding to the number of origins, the specific values in each sub-row indicate the number of Found (resp. Not Found) paths by the respective approaches. As mentioned earlier, a time limit determines the termination of RWR. In our experiments, we triple the maximum/worst searching time of Dijkstra and Greedy-Dijkstra algorithms, and use it as the terminate condition for RWR. As can be seen, Greedy-Dijkstra strategy outperforms the other two algorithms in all settings. However, on the opposite, RWR always find the least number of demanded paths even if it was granted three times longer of the searching time. When the number of origins increases, Dijkstra and the greedy variant are capable to discover more satisfied paths since the more origins, the greater flexibility of initial locations we have. Besides, another interesting observation from Table. 1 is the relationship between the number of origins and the searching time. We can discover from the last column that for RWR, the number of found paths drops when the number of origins increases from 200 to 576, which is different from the other two algorithms. The reason is that plenty of origin locations allows Dijkstra and the greedy variant quickly discover a valid path and hence the time limit for RWR will be shortened. Last note we want to emphasis is that those cases which do not find any satisfied path might be due to the nonexistence of solution at all – e.g., find a path with 2 museums and 3 parks within 500 meters.

4 CONCLUSION

We introduced a novel query – $\mathbf{SP}_O^R$, which returns not only a semantically diverse path along which the POIs fail within a certain cumulative distance range, but also considers constrains on the set of possible origin-locations. $\mathbf{SP}_O^R$ enables tourists to concomitantly select a hotel and obtain a customized path for visiting POIs.

We note that we constructed an actual POI network based on real data, which can be useful for the further research since, at the time of this work, there is no online available resource of such data. We integrated OpenStreetMap and TripAdvisor to generate our datasets.

There are several directions of our future work. First and foremost, we would like to extend the definition of $\mathbf{SP}_O^R$ so that it can incorporate well-defined preferences and/or optimization criteria. Concomitantly, we will explore the development of novel data structures that would enable more efficient processing of $\mathbf{SP}_O^R$. Complementary to these, we would like to incorporate the time-dependency when catering to the travel constraints/budget, to capture the fact that different starting times may impact the generated paths corresponding to a solution of $\mathbf{SP}_O^R$.

ACKNOWLEDGMENTS

The work on this research project has been partially supported by the NSF SWIFT grant 2030249 and NSF CCF grant 1637541.

REFERENCES

[1] Jie Bao and Yu Zheng. 2017. *Location-Based Recommendation Systems.*
[2] Jie Bao, Yu Zheng, David Wilkie, and Mohamed F. Mokbel. 2015. Recommendations in location-based social networks: a survey. *GeoInformatica* 19, 3 (2015).
[3] B. Wayne Bequette. 2001. *Process Control: Analysis, Design, Simulation.* Prentice Hall.
[4] Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. 2005. On map-matching vehicle tracking data. In *VLDB.*
[5] Lisi Chen, Shuo Shang, Christian S. Jensen, Bin Yao, and Panos Kalnis. 2020. Parallel Semantic Trajectory Similarity Join. In *IEEE ICDE.*
[6] EW Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.
[7] Yunjun Gao, Baihua Zheng, Gencai Chen, and Qing Li. 2010. Algorithms for constrained *k*-nearest neighbor queries over moving object trajectories. *GeoInformatica* 14, 2 (2010).
[8] Haosheng Huang, Georg Gartner, Jukka Matthias Krisp, Martin Raubal, and Nico Van de Weghe. 2018. Location based services: ongoing evolution and research agenda. *J. Locat. Based Serv.* 12, 2 (2018).
[9] Richard M Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations.*
[10] Amgad Madkour, Walid G. Aref, Faizan Ur Rehman, Mohamed Abdur Rahman, and Saleh M. Basalamah. 2017. A Survey of Shortest-Path Algorithms. *CoRR* abs/1705.02044 (2017).
[11] Harvey J. Miller. 2017. *Location-Aware Technologies.*
[12] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, et al. 2013. Semantic trajectories modeling and analysis. *ACM (CSUR)* 45, 4 (2013).
[13] Mahsa Pourbafrani, Shuai Jiao, and Wil M. P. van der Aalst. 2021. SIMPT: Process Improvement Using Interactive Simulation of Time-Aware Process Trees. In *RCIS.*
[14] Xiaozhao Song, Jiajie Xu, Rui Zhou, Chengfei Liu, Kai Zheng, Pengpeng Zhao, and Nickolas Falkner. 2020. Collective spatial keyword search on activity trajectories. *GeoInformatica* 24, 1 (2020), 61–84.
[15] Xu Teng, Goce Trajcevski, Joon-Seok Kim, and Andreas Züfle. 2020. Semantically Diverse Path Search. In *IEEE MDM.*
[16] Xu Teng, Jingchao Yang, Joon-Seok Kim, Goce Trajcevski, Andreas Züfle, and Mario A Nascimento. 2019. Fine-Grained Diversification of Proximity Constrained Queries on Road Networks. In *SSTD.*
[17] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *ICDM.*
[18] Fan Zhou, Hantao Wu, Goce Trajcevski, Ashfaq A. Khokhar, and Kunpeng Zhang. 2020. Semi-supervised Trajectory Understanding with POI Attention for End-to-End Trip Recommendation. *ACM Trans. Spatial Algorithms Syst.* 6, 2 (2020).
[19] Fan Zhou, Ruiyang Yin, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Jin Wu. 2019. Adversarial Point-of-Interest Recommendation. In *WWW.*