Searching Trajectories by Regions of Interest

Searching Trajectories by Regions of Interest

(Extended Abstract)

Shuo Shang[1]   Lisi Chen[2]   Christian S. Jensen[3]   Ji-Rong Wen[4]   Panos Kalnis[5]

[1,5]CEMSE, King Abdullah University of Science and Technology, Saudi Arabia
[2]Wollongong University, Australia
[3]Department of Computer Science, Aalborg University, Denmark
[4]School of Information, Renmin University, China
[1]jedi.shang@gmail.com   [2]chenlisi.cs@gmail.com   [3]csj@cs.aau.dk   [4]jrwen@ruc.edu.cn   [5]panos.kalnis@kaust.edu.sa

I. INTRODUCTION

We consider a setting that encompasses a set of spatial trajectories as well as a set of spatial point objects, which can be geo-tagged photos, geo-tagged tweets, or points of interest such tourist attractions, shops, and restaurants. In this setting, we propose and study a novel query type called trajectory search by regions of interest (TSR query). This query takes the set of trajectories as its argument, and it takes a set of regions of interest, that each contains spatial point objects, as a parameter. The query returns the trajectories with the highest spatial-density correlation to the query regions. This type of query is intended as a foundation for enabling new location-based services, including novel trip planning and recommendation applications.

In most existing studies on trajectory search [1], the query parameter is a set or sequence of point locations. However, in some cases, users may be interested in regions that contain points of interest, e.g., a scenic area, a commercial district, or a dining area, that contain points of interest (POIs), geo-tagged photos, tweets, or reviews. Moreover, when planning a trip in unfamiliar surroundings, users may fail to specify intended locations exactly and may use intended regions instead. These two common cases motivate our study. To the best of our knowledge, this is the first study of region-based trajectory search in spatial networks that takes spatial-object density into account. Previous studies [1] use spatial distance as the sole criterion when computing results. However, spatial distance in itself fails to fully capture the relationship between a trajectory and a set of regions. For example, a user may not be satisfied with a travel route with a relatively sparse distribution of nearby objects, although the route is spatially close to the centers of the given query regions.

This is illustrated in Figure 1, where c_1, c_2, and c_3 are TSR query regions, p_1, p_2, and p_3 are the corresponding region centers, and r_1, r_2, and r_3 are the radiuses. Points $p_3, p_4, ..., p_8$ are sample points in trajectories. Among the sample points in τ_1, p_6, p_7, and p_8 are the sample points that are closest to centers p_1, p_2, and p_3. Among the sample points in τ_2, p_4 and p_5 are the sample points that are closest to centers p_1 and p_2. Each region contains several spatial objects. If only spatial proximity to the region centers is considered [1], trajectory τ_2 is returned because τ_2 is spatially closest to the region centers. However, if we consider the distributions of the spatial objects, τ_2 is less attractive than τ_1 because it is further away from
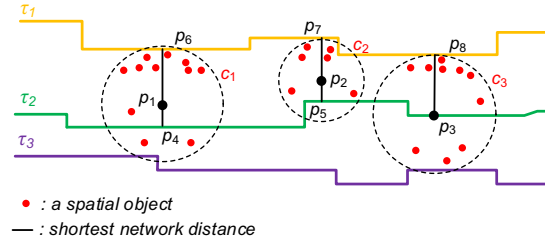

Fig. 1.   An example of the TSR query

the point objects contained in the regions. When taking both spatial distances to the objects in the query regions and spatial-object density into account, τ_1 is the best choice for trajectory recommendation.

TSR query processing faces three challenges: how to model the spatial-density correlation between regions and trajectories, how to effectively prune the search space during query processing, and how to effectively schedule multiple so-called query sources. To tackle these challenges, new metrics are defined to model spatial-density correlations. An efficient trajectory search algorithm is developed that exploits upper and lower bounds to prune the search space and that adopts a query-source selection strategy, as well as integrates a heuristic search strategy based on priority ranking to schedule multiple query sources. The performance of TSR query processing is studied in extensive experiments based on real and synthetic spatial data. To the best of our knowledge, no existing method can compute the TSR query efficiently.

II. SOLUTION OVERVIEW

Baseline Algorithm. The uniform-speed search (USS) is a straightforward approach to computing the TRS query based on the filter-and-refine paradigm. Given a trajectory data set T and a query region set C, USS asks for the trajectories spatially close to the spatial-object dense areas within the query regions. Each region center $c_i.m$ ($c_i \in C$) is selected as a so-called query source, and network expansions (i.e., Dijkstra's expansion [2]) are performed from the query sources at the same speed to explore the network. A pair of upper and lower bounds of spatial-density correlation are defined to prune the search space. By integrating the results, the trajectory with the maximum spatial-density correlation to C is found.

The uniform-speed search (USS) lacks an effective scheduling strategy for multiple query sources, which may lead to

IEEE
computer
society

poor performance. In addition, if two query region centers are close to each other and both are selected as query sources, their search spaces may overlap substantially, which again decreases performance.

Best-Expansion Search Algorithm. To achieve better performance, we develop a best-expansion search (BES) algorithm. First, we reuse an existing query-source selection strategy [4] to select a set of query sources from the set of query region centers. Second, we define new upper and lower bounds on the spatial-density correlation to enable pruning. Third, we propose a method to schedule expansion from the query sources effectively. Specifically, we establish and maintain a dynamic priority ranking heap during query processing. At each step, we expand from the top-ranked query source until a new top-ranked query source appears. The BES algorithm has two major advantages over USS: (i) it further prunes the search space to avoid overlapping traversals; (ii) it is able to focus on trajectories more likely to be the solution.

Complexity Analysis. The BES algorithm follows the filter-and-refine paradigm. In the filtering phase, it performs x Dijkstra's expansions, yielding a time complexity of $O(x(|V|\log|V| + |E|))$, where $x \leq |C|$ is the (constant) number of query sources, and $|V|$ and $|E|$ are the vertex and edge counts in graph G. The filtering phase of USS has a time complexity of $O(|C|(|V|\log|V| + |E|))$. In the refinement phase, we perform x Dijkstra's expansions to refine each trajectory in candidate set T'_f, which has a time complexity of $O(x|T'_f|(|V|\log|V| + |E|))$. The refinement phase of USS has a time complexity of $O(|C||T_f|(|V|\log|V| + |E|))$, where $|T'_f| \leq |T_f|$. When combining the two phases, the time complexity of BES is $O(x(|V|\log|V| + |E|)) + O(x|T'_f|(|V|\log|V| + |E|)) = O(|T|(|V|\log|V| + |E|))$. In the worst case, the time complexity of BES is the same as that of USS. However, in experiments on real data sets, BES outperforms USS by a factor of 2–3 due to its pruning capabilities.

Extension. It is also possible that a traveler may specify a visiting sequence for intended regions (e.g., $c_1 \to c_2 \to c_3$ is the intended visiting sequence of three regions). The USS and BES algorithms are extended to efficiently process TSR queries with a sequence.

III. Experiments

We evaluate the performance of USS and BES algorithms and the BES algorithm without heuristic scheduling (denoted by BES-w/o-h) on the Beijing Road Network (BRN), and the North America Road Network (NRN)[1]. For BRN, we use a real taxi trajectory data set and a real data set of points of interest (spatial objects), which contain 800,000 trajectories and 300,000 POIs. For NRN, larger synthetic data is used to study scalability. NRN is associated with 4,000,000 trajectories and 1,800,000 spatial objects.

Figure 2 presents the performance of the algorithms when varying the number of trajectories $|T|$. Intuitively, a larger $|T|$ causes more trajectories to be processed and yields a larger search space. As a result, both the CPU time and the

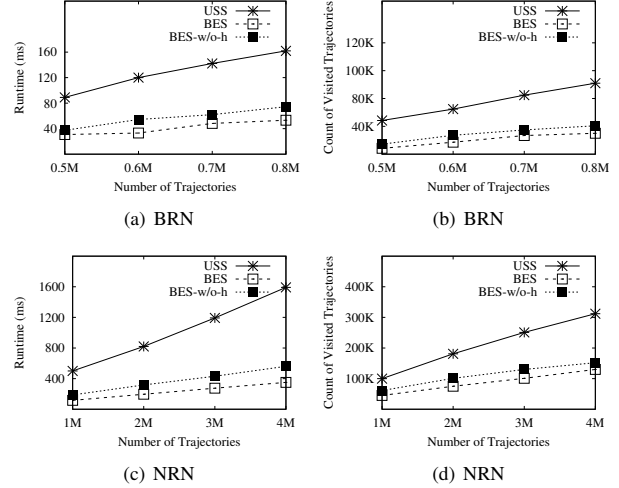[1]http://www.cs.utah.edu/~lifeifei/SpatialDataset.htm

Fig. 2. Effect of the number of trajectories

count of visited trajectories are expected to be higher for all three algorithms. However, Figure 2 shows that BES is able to outperform USS by almost a factor of 8 on NRN (for both CPU time and visited trajectories). The upper and lower bounds based on the query-source selection strategy can improve the efficiency of USS by a factor of 4–6, and the heuristic scheduling strategy can further improve the efficiency by a factor of 2, in terms of both CPU time and the count of visited vertices.

IV. Conclusion

We propose and study a novel problem, namely trajectory search by regions of interest (TSR query), that finds the trajectory with the highest spatial-density correlation to a set or sequence of query regions. This type of functionality targets novel trip planning and recommendation applications as well as location-based services more generally. To compute the TSR query efficiently, we develop a best-expansion search algorithm that exploits upper and lower bounds to prune the search space and that adopts a query-source selection strategy, as well as a heuristic search strategy based on priority ranking to schedule query sources. The performance of the TSR query was investigated through extensive experiments on both real and synthetic spatial data. For a comprehensive coverage, see the full version of the paper [3].

References

[1] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie. Searching trajectories by locations: an efficiency study. In *SIGMOD*, pages 255–266, 2010.

[2] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math*, 1:269–271, 1959.

[3] S. Shang, L. Chen, C. S. Jensen, J. Wen, and P. Kalnis. Searching trajectories by regions of interest. *IEEE Trans. Knowl. Data Eng.*, 29(7):1549–1562, 2017.

[4] S. Shang, K. Zheng, C. S. Jensen, B. Yang, P. Kalnis, G. Li, and J. Wen. Discovery of path nearby clusters in spatial networks. *IEEE Trans. Knowl. Data Eng.*, 27(6):1505–1518, 2015.