Finding Most Popular Indoor Semantic Locations Using Uncertain Mobility Data

(Extended Abstract)

Huan Li[†] Hua Lu[‡] Lidan Shou[†] Gang Chen[†] Ke Chen[†]

[†] Department of Computer Science, Zhejiang University, China

[‡]Department of Computer Science, Aalborg University, Denmark

{lihuancs, should, cg, chenk}@zju.edu.cn, luhua@cs.aau.dk

I. Background And Motivation

In recent years, people's indoor movements are increasingly datafied due to the rapid deployment of indoor location-based service infrastructures. Akin to what has been done using GPS data [1], proper analysis on indoor mobility data can reveal insights that are otherwise difficult to obtain. As a typical example, many applications can benefit from analyzing the *indoor flows* that disclose the number of people passing by particular indoor regions during a past time interval. Examples include exhibition planning, location-based advertising, etc.

In this study, we formulate the problem of finding top-k popular *indoor semantic locations* with the highest flows during a past time interval. In our setting, the mobility information of an object at a time t is captured by a set of probabilistic samples in the format of $(loc, prob)$. Such a format is often seen in indoor positioning services based on wireless infrastructure. The problem is nontrivial due to two challenges. The first challenge is the difficulty in obtaining *reliable* flow values due to the inherent uncertainty in multiple location samples reported at discrete timestamps. The data uncertainty together with complex indoor topology entails an appropriate formulation of indoor flows. The second challenge comes from the heavy computational workloads on the samples for large numbers of indoor objects. As the existing approaches [1], [4] are unsuitable for such complex indoor settings, we propose a number of novel techniques in our full paper [2] to address these challenges. Our key contributions are as follows.

- We formulate the indoor flow definition and design a complete set of techniques for efficiently computing the flows for individual indoor semantic locations.
- We design search algorithms for finding top-k indoor popular semantic locations with highest flows.
- We conduct extensive experiments to verify the efficiency, scalability, and effectiveness of our approach.

II. Problem Formulation

In our definition, *semantic locations* (S-locations) refer to the regions relevant to applications, e.g., a shop in a mall. *Positioning locations* (P-locations) refer to the points returned by a positioning system. We further distinguish them into two subclasses. A set of *partitioning P-locations* partition space into *cells* in that an object cannot move from one to another without passing any of these P-locations. In contrast, *presence P-locations* only imply the presence of a positioned object.

Specifically, the positioning record (o, X, t) is reported to an *Indoor Uncertain Positioning Table* (IUPT) non-periodically, meaning that the object o's location at time t is described by a sample set X. Each sample $e(loc, prob)$ in X means that o is at a P-location loc with probability $prob$.

In order to know how many objects appeared in a given S-location q during a past time interval $[t_s, t_e]$, we first define the uncertainty-aware *object presence* as follows. We obtain an object o's sample sets sequence $\mathcal{X} = (X_1, \dots, X_n)$ and consider its possible paths in the Cartesian product of the relevant P-location sets. The probability of each possible path $\phi_i = (loc^i_1, \dots, loc^i_n)$ is computed as $pr_i = \prod_{1 \leq j \leq n} prob^i_j$ where $prob^i_j$ is the probability associated with P-location loc^i_j in sample set X_j. Introducing the *pass probability* [1] $pr_{\phi_i \rightsquigarrow q}$, we compute an object o's *presence* in q as

$$\Phi_{t_s,t_e}(q, o) = \frac{\sum_{\phi_i \in P}(pr_{\phi_i \rightsquigarrow q} \cdot pr_i)}{\sum_{\phi_i \in P} pr_i} \qquad (1)$$

Definition 1 (Indoor Flow): Given an S-location q, a set O of indoor moving objects, and a time interval $[t_s, t_e]$, the indoor flow for q is $\Theta_{t_s,t_e,O}(q) = \sum_{o \in O} \Phi_{t_s,t_e}(q, o)$.

Problem 1 (Top-k Popular Location Query, TkPLQ): Given a set Q of indoor semantic locations, an IUPT for a set O of indoor moving objects, and a time interval $[t_s, t_e]$, an indoor top-k popular location query returns k S-locations in a k-subset $Q_k \subseteq Q$ such that $\forall q \in Q_k, \forall q' \in Q \setminus Q_k, \Theta_{t_s,t_e,O}(q) \geq \Theta_{t_s,t_e,O}(q')$.

III. Algorithms for TkPLQ

We first design data structures that facilitate data access in flow computing. We also design a data reduction method that reduces the possible indoor paths to consider. Based on these techniques, we propose search algorithms for TkPLQ.

A. Data Structures and Data Reduction Method

The problem definition involves possible paths that consist of P-locations, whereas the query set is defined on S-locations (or say, the parent cells). To bridge this gap, we devise an *indoor space location graph* G_{ISL} that captures the topological relationship between them. As shown in Fig. 1, a cell c_1 consists of two S-locations, i.e., rooms r_1 and r_2; it contains

[1] We compute the pass probability that a path ϕ passes q as one minus the probability that none of the consecutive P-location pairs in ϕ passes q.

a presence P-location p_7 and is divided by two partitioning P-locations p_4 and p_9.

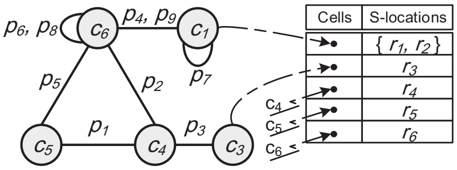

Fig. 1. Indoor Space Location Graph G_{ISL}

Referring to G_{ISL}, we build an *indoor location matrix* M_{IL} for quickly searching relevant cells (S-locations) of two sequential P-locations in a path. E.g., $M_{IL}[p_4, p_9] = \{c_1, c_6\}$, if we see a path involving p_4 and p_9 sequentially, we can tell that the object is in either cell c_1 or c_6. Also, $M_{IL}[p_8, p_8] = c_6$ indicates that a path containing p_8 should pass through cell c_6.

Given a positioning sequence $\mathcal{X} = (X_1, \ldots, X_n)$, the *maximum* number of possible paths formed by Cartesian product is as large as $\prod_{1 \le i \le n} |X_i|$. We bring up two operations to reduce the number. For each set X_i, we use an *intra-merge* to combine the samples from the P-locations that are logically equivalent in constructing M_{IL} (e.g., p_6 and p_8). Also, we use an *inter-merge* to compress the sequence length $|\mathcal{X}|$ by merging the consecutive sets that contain the identical P-locations.

B. Flow Computing and TkPLQ Search

We first introduce our flow computing method for individual S-locations. Given an S-location q, we fetch and go through all relevant positioning records within $[t_s, t_e]$. We use an 1DR-tree [3] to index the IUPT on its time attribute. The matrix M_{IL} is checked to determine if the current path to be generated is valid, and only the valid ones will be involved in subsequent path generation. Afterward, we compute the object presences according to Definition 1 and adds them to q's overall flow.

We proceed to present different TkPLQ search algorithms. As shown in Fig. 2, a Naive algorithm sorts the top-k results after blindly computing each individual query location's flow. To improve its efficiency, a Nested-Loop algorithm caches each encountered object's presences to avoid re-computation. We further devise a Best-First algorithm that gives priority to a set of promising query locations with greater flow overestimates. To quickly locate those promising locations and their relevant object samples, we carry out a join of a query location R-tree and an object COUNT-aggregate R-tree [5]. Readers can refer to [2] for the details of our presented algorithms.

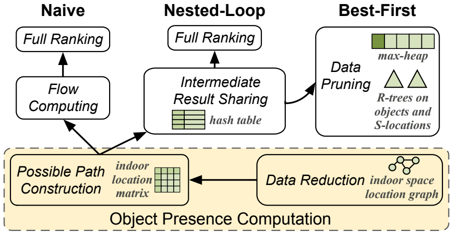
Naive
Nested-Loop
Best-First
Full Ranking
Full Ranking
Flow Computing
Intermediate Result Sharing
hash table
Data Pruning
max-heap
R-trees on objects and S-locations
Possible Path Construction
indoor location matrix
Data Reduction
indoor space location graph
Object Presence Computation

Fig. 2. Naive, Nested-Loop, and Best-First Algorithms for TkPLQ

IV. Experimental Results

We compare our algorithms Naive, NL (Nested-Loop) and BF (Best-First) to several alternatives. Among them, SC (simple counting) method picks the sample with the highest probability and adds 1 to all its containing S-locations' flow values; SC-ρ differs from SC only in that it picks all the samples whose probability exceeds a threshold ρ; MC (Monte Carlo) method executes a certain number of simulations, in each of which all the positioning records are sampled to be certain. As a result, the top-k locations are ranked based on their average flows in all the simulations.

We study both the top-k search efficiency and effectiveness. For the former, we consider the average running time and *pruning ratio*; for the latter, we measure the *recall* and *Kendall coefficient* τ with respect to the ground truth. In a default query setting, the best performance results on the real data collected from a Wi-Fi based positioning system are reported in Table I.

TABLE I
Performance Comparison in Default Setting

Methods	Running time (sec.)	Pruning ratio (%)	Kendall coefficient	Recall (%)
SC	**0.6**	-	0.007	62.2
SC-ρ ($\rho = 0.25$)	1.1	-	0.382	75.6
MC, *900 rounds*	1.7×10^4	-	0.712	86.7
BF	4.4	**59.4**	0.859	93.3
NL	9.5	19.2	same as above.	
Naive	59.1	19.2	same as above.	

In general, SC and SC-ρ incur short time costs but yield very poor effectiveness; MC that uses simulations incurs extremely long running time. In contrast, by applying our uncertainty-aware flow computing, BF and NL's effectiveness measures are significantly higher; BF also achieves a good balance between the efficiency and effectiveness.

With a large synthetic dataset, we also test the effect of the data uncertainty related to the *maximum positioning period* T and the *indoor positioning error* μ. The results are reported in Fig. 3. A larger T makes the location updates less frequent, which causes the data uncertainty to increase and the query result quality to degrade. Nevertheless, BF still outperforms the best; its τ keeps above 0.77 in all tests. When μ increases, both SC and SC-ρ's τ decrease clearly as these methods counting on the positioning records are very sensitive to the positioning errors. Still, BF outperforms MC as BF considers the valid possible paths thoroughly on the uncertain positioning data. When $\mu = 7m$, its τ is still higher than 0.77.

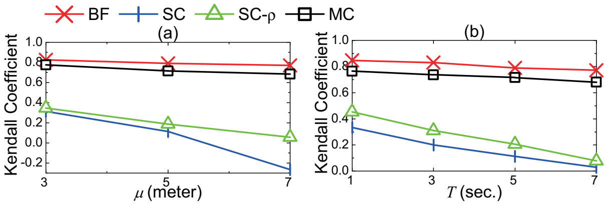

Fig. 3. Kendall Coefficient vs. T on Synthetic Data

References

[1] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from GPS data. *PVLDB*, 3(1): 1009–1020, 2010.

[2] H. Li, H. Lu, L. Shou, G. Chen, and K. Chen. Finding Most Popular Indoor Semantic Locations Using Uncertain Mobility Data. *IEEE Trans. Knowl. Data Eng.*, 2018.

[3] H. Lu, B. Yang, and C. S. Jensen. Spatio-temporal joins on symbolic indoor tracking data. In *ICDE*, pp. 816–827, 2011.

[4] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio temporal aggregation using sketches. In *ICDE*, pp. 214–225, 2004.

[5] Y. Tao and D. Papadias. Range aggregate processing in spatial databases. *IEEE Trans. Knowl. Data Eng.*, 16(12): 1555–1570, 2004.