Finding Most Popular Indoor Semantic Locations Using Uncertain Mobility Data

†Huan Li, ‡Hua Lu, †Lidan Shou, †Gang Chen, and †Ke Chen

†College of Computer Science and Technology, Zhejiang University, China
‡Department of Computer Science, Aalborg University, Denmark

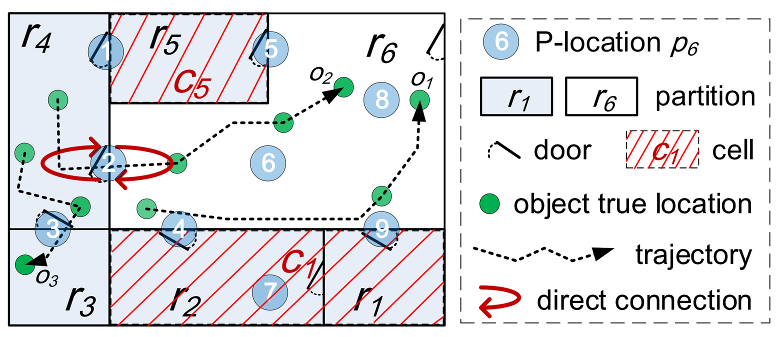
1. Introduction

- Indoor movements are increasingly datafied due to the rapid growth of indoor LBS infrastructures. Proper analysis can reveal insights that are otherwise difficult to obtain.
  - **Indoor flow analysis.** the number of people passing by particular indoor regions during a past time interval. Application include exhibition planning, location-based advertising, etc.
- The problem of finding the top-k popular *indoor semantic locations* with the highest flows during a past time interval.
  - The mobility information of an object at a time t is captured by a set of *probabilistic samples* in the format of $(loc, prob)$.
  - **The first challenge** is the difficulty in obtaining *reliable* flow values due to the inherent uncertainty in multiple 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.
- A complete set of novel techniques for indoor flow analysis.
  - We formulate the definition of *indoor flows* by taking into account both data uncertainty and indoor topology.
  - We devise *data structures* to facilitate accessing the data relevant to flow computing, and a *data reduction method* to significantly reduce the intermediate data to be processed.
  - We design *search algorithms* for finding indoor top-k popular locations.

2. Problem Formulation

- ***Semantic locations*** (S-locations) refer to regions relevant to applications, e.g., a shop.
- ***Positioning locations*** (P-locations) refer to points returned by indoor positioning system.
  - **Partitioning P-locations** partition space into *cells* in that objects cannot move from one to another without passing these P-locations. **Presence P-locations** only imply the presence of a positioned object.
- A record (o, X, t) is reported to an ***Indoor Uncertain Positioning Table*** non-periodically, meaning o's location at 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 $prob$.
- ***Uncertainty-aware object presence*** in a S-location q during time interval $[t_s, t_e]$.
  - For each object o's sample sets sequence $(X_1, \dots, X_n)$ ➔ Obtain possible paths in the *Cartesian product* $\phi_i = (loc^i_1, \dots, loc^i_n)$ ➔ Compute path probability as $pr_i = \prod_{1\le j\le n} prob^i_j$ where $prob^i_j$ is the probability associated with P-location loc^i_j in X_j.
  - The *pass probability* that ϕ passes q is 1 minus the probability that none of consecutive P-location pairs in ϕ passes q ➔ o's *presence* in q is $\Phi_{t_s,t_e}(q, o) = \frac{\sum_{\phi_i\in P}(pr_{\phi_i\rightsquigarrow q}\cdot pr_i)}{\sum_{\phi_i\in P} pr_i}$.
- ***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 $\Theta_{t_s,t_e,O}(q) = \sum_{o\in O}\Phi_{t_s,t_e}(q, o)$.
- ***Top-k Popular Location Query, TkPLQ.*** Given a set Q of indoor semantic locations, 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 $\forall q \in Q_k, \forall q' \in Q \setminus Q_k, \Theta_{t_s,t_e,O}(q) \ge \Theta_{t_s,t_e,O}(q')$.

A running example



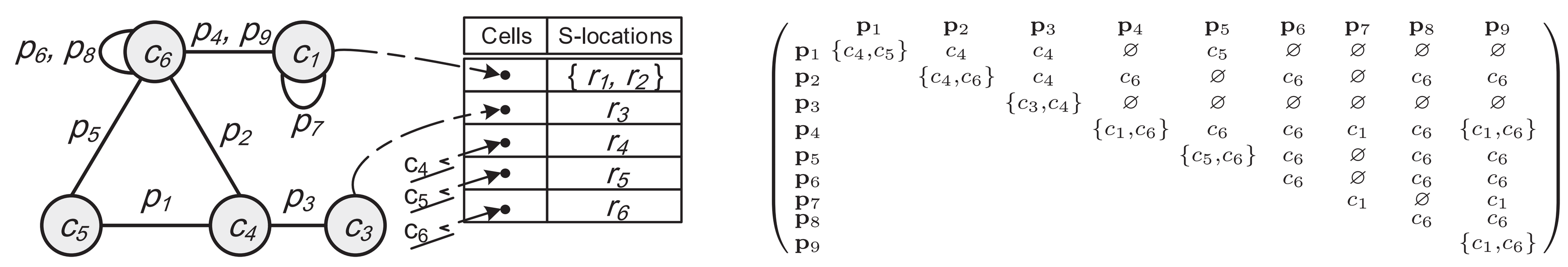
oid , *X* , *t*	*oid* , *X* , *t*
o_1 , $\{(p_4, 1.0)\}$, t_1	o_1 , $\{(p_8, 1.0)\}$, t_4
o_2 , $\{(p_1, 0.5), (p_2, 0.5)\}$, t_1	o_2 , $\{(p_5, 0.3), (p_6, 0.6), (p_8, 0.1)\}$, t_5
o_3 , $\{(p_2, 0.6), (p_3, 0.4)\}$, t_2	o_3 , $\{(p_2, 0.4), (p_3, 0.6)\}$, t_5
o_1 , $\{(p_9, 1.0)\}$, t_3	o_2 , $\{(p_5, 0.2), (p_6, 0.3), (p_8, 0.5)\}$, t_6
o_2 , $\{(p_2, 0.7), (p_4, 0.3)\}$, t_3	o_3 , $\{(p_3, 1.0)\}$, t_8

- An object o_3 has 4 possible paths during $[t_1, t_8]$, i.e., $\phi_1 = (p_2, p_2, p_3)$, $\phi_2 = (p_2, p_3, p_3)$, $\phi_3 = (p_3, p_2, p_3)$ and $\phi_4 = (p_3, p_3, p_3)$. In particular, ϕ_1's probability is $0.6 \times 0.4 \times 1.0 = 0.24$.
- The possible path ϕ_1 contains sequential P-location pairs (p_2, p_2) and (p_2, p_3). For (p_2, p_2), we find two direct connections, and have $pr_{p_2,p_2\rightsquigarrow r_6} = pr_{p_2,p_2\rightsquigarrow r_4} = 1/2$. Likewise, for pair (p_2, p_3), $pr_{p_2,p_3\rightsquigarrow r_4} = 1$ and $pr_{p_2,p_3\rightsquigarrow r_6} = 0$. The pass probability $pr_{\phi_1\rightsquigarrow r_6} = 1 - (1 - 1/2)\cdot(1 - 0) = 0.5$.
- The presence $\Phi_{t_1,t_8}(r_6, o_3) = 0.5 \cdot 0.24 = 0.12$, and $\Phi_{t_1,t_8}(r_1, o_3) = 0$.
- S-location r_6's indoor flow is $\Theta_{t_1,t_8,O}(r_6) = \sum_{1\le i\le 3}\Phi_{t_1,t_8}(r_6, o_i) = 1+0.85+0.12 = 1.97$, r_1's is $\Theta_{t_1,t_8,O}(r_1) = \sum_{1\le i\le 3}\Phi_{t_1,t_8}(r_1, o_i) = 0.5 + 0 + 0 = 0.5$. A T1PLQ during $[t_1, t_8]$ returns room r_6.

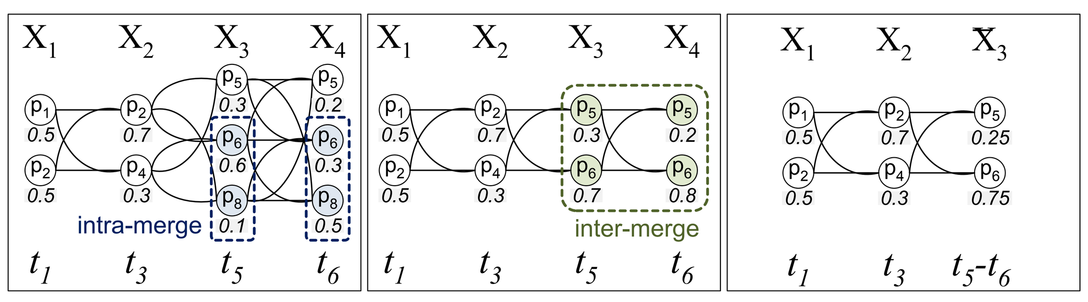
3. Algorithms for TkPLQ

3.1 Data structures and data reduction method.

- To bridge the gap between P-locations and S-locations, we devise an ***indoor space location graph***. A cell c_1 ➔ rooms r_1 and r_2 ➔ partitioning P-locations p_4 and p_9.
- We further build an ***indoor location matrix*** M_{IL} for quickly searching relevant cells (S-locations) of two sequential P-locations in a path. $M_{IL}[p_4, p_9] = \{c_1, c_6\}$ and $M_{IL}[p_8, p_8] = c_6$.

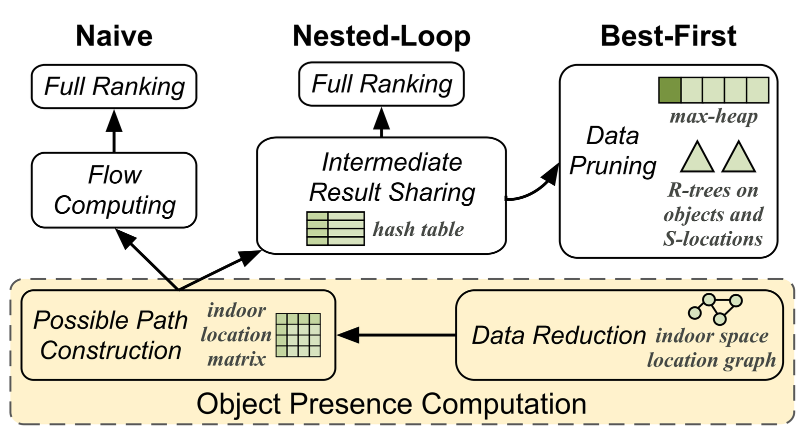


- For sequence $(X_1, \dots, X_n)$, the *maximum* number of possible paths is as large as $\prod_{1\le i\le n}|X_i|$.
  - For each set X_i, we use an ***intra-merge*** operation to combine the samples from the P-locations that are logically equivalent in constructing M_{IL} (e.g., p_6 and p_8).
  - We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



3.2 Flow computing and TkPLQ search.

- Flow computing for individual an S-location ☛ fetch and go through all relevant positioning records within $[t_s, t_e]$ that are indexed by an 1DR-tree.
  - 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.
- **Naive algorithm** sorts top-k results after blindly computing each query location's flow.
- **Nested-Loop** caches each encountered object's presences to avoid re-computation.
- **Best-First** gives priority to those promising query locations with greater flow overestimates. To quickly locate the relevant object samples, we carry out a join of a query location R-tree and an object COUNT-aggregate R-tree.



4. Experimental Results

- We compare Naive, Nested-Loop and Best-First to several alternatives.
  - 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.

4.1 Performance comparisons using a real-world dataset.

- **Efficiency metrics** ☛ Average running time and *Pruning ratio*; **Effectiveness metrics** ☛ *Recall* and *Kendall coefficient* τ w.r.t the ground truth.
- SC and SC-ρ incur short time costs but yield very poor effectiveness; MC that uses simulations incurs extremely long running time.
- By applying uncertainty-aware flow computing, BF and NL's effectiveness measures are significantly higher; BF achieves a good balance between efficiency and effectiveness.

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.	

4.2 Studies on data uncertainty using a synthetic dataset.

- A larger T (*maximum positioning period*) makes location updates less frequent, which causes data uncertainty to increase and query result quality to degrade. BF still outperforms the best; its τ keeps above 0.77 in all tests.
- When *indoor positioning error* μ increases, SC and SC-ρ's τ decrease clearly as they are sensitive to data errors. Still, BF outperforms MC as BF considers valid possible paths thoroughly on uncertain positioning data.