On Publication of the Research

Dr. Yu Zheng

Researcher @ Web Search & Mining Group
Microsoft Research Asia

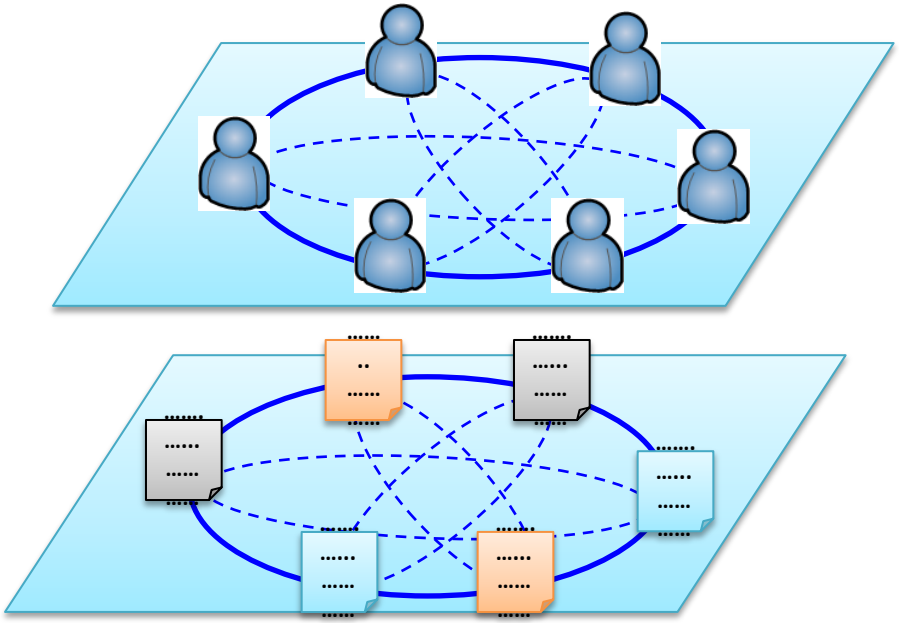
http://research.microsoft.com/en-us/people/yuzheng/

A reminder

- Do not treat my experiences as rules
- Do better using your intelligence and creativity

Outline

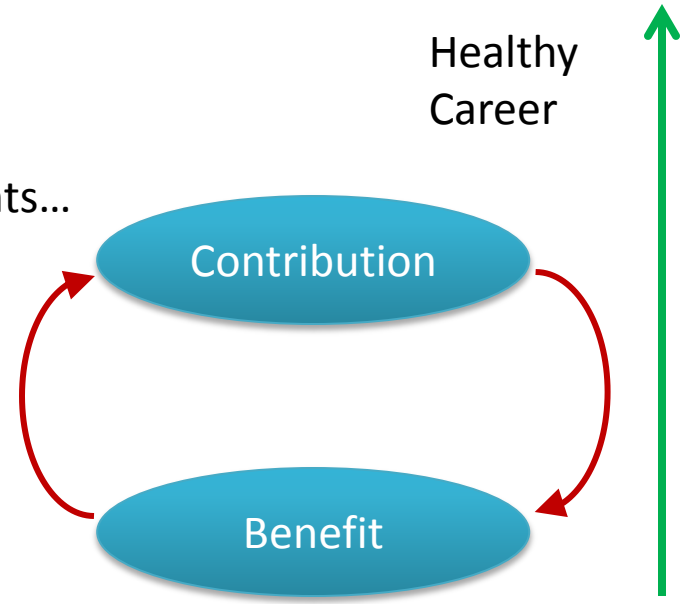
- Why publishing a research paper
- Category of research publications
- Styles of different conferences
- How to evaluate a research paper
- Art of the writing
- Warnings



Why publishing a research paper

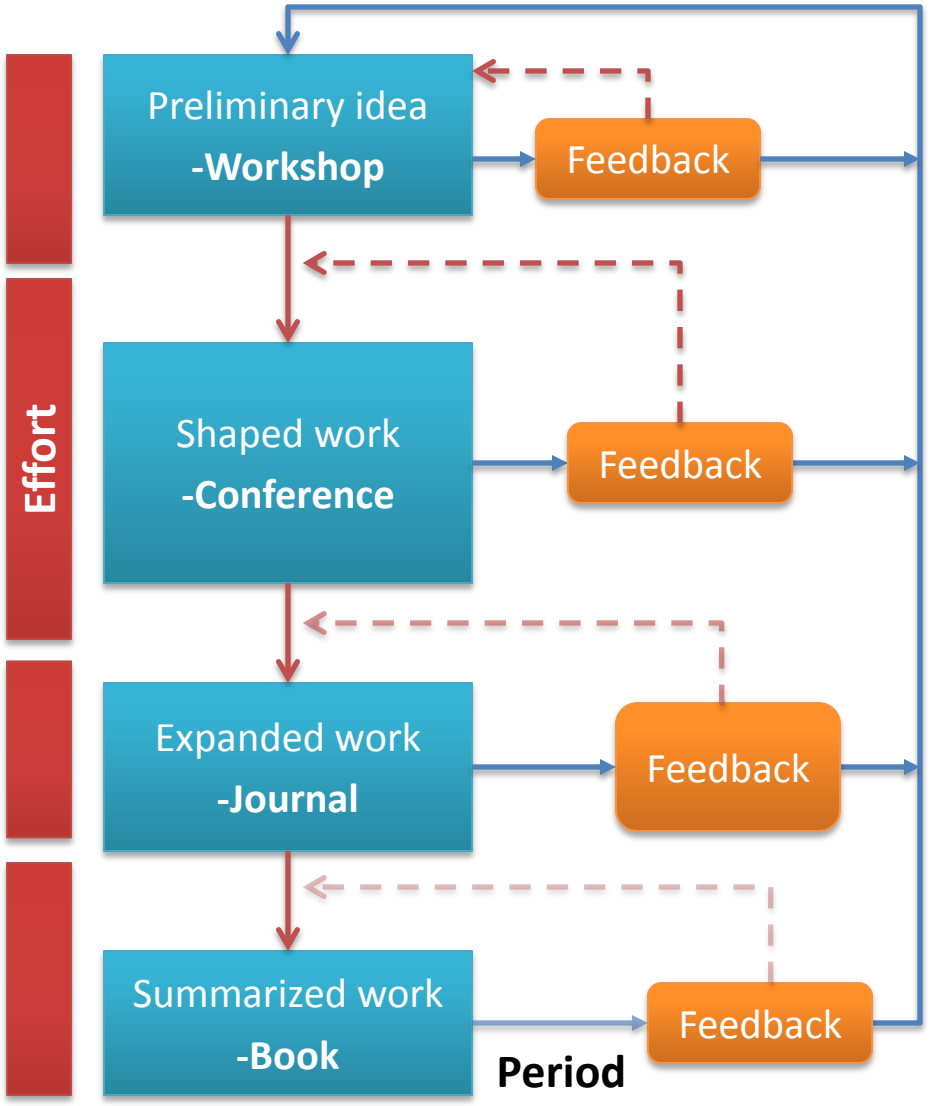
- Contribute to the community
  - Propose new problem and research directions
  - Share your idea and methodologies to others
  - Feedback and improve other people's work
  - Educate and knowledge new-comers to the community

- Benefit from publications
  - Well document your work, shape your thoughts…
  - Practice presentation skill
  - Get feedback from others
  - Connect to peers and identify collaboration
  - Build personal credit in the community
  - Respect, honor, and awards……



Category of research publications

- Technical report
- Workshop publications
- Conference publications
  - Full/oral papers
  - Short paper
  - Posters
  - Demos
  - Videos
  - Doctoral Colloquium
  - On the progress work
  - Tutorial
- Journal publications
  - Full research articles
    - Regular
    - Special issues
  - Review and comment papers
- Book



Styles of different conferences

- Human and Interfaces
  - CHI, UIST, …
- Applications and systems
  - WWW, SIGIR, …
  - Ubicomp, Pervasive, Percom…
  - KDD industrial track
  - Mobisys, ACM MM…
- Theory and models
  - SIGMOD, VLDB, ICDE, KDD research track…
  - AAAI, IJCAI…
  - ICML, NIPS…

	Human and interaction	Applications and systems	Theory and models
Language style	Natural	Natural + Formal	Formal
Scales of Experiment	Rich	Middle	Small
Experiment styles	User study/in the field study	Synthetic test/lab. test /user study	Synthetic test/lab. test
Lemma and proof	Rare	Middle	Rich

How to evaluate a conference paper

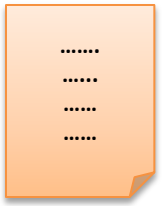
- Relevance to a conference or journal
- Novelty and originality
- Significant contribution
  - **Insight**
  - Strong points
- Technical sound
  - Methodology
  - experiments
- Presentation
  - Clear structure
  - Good readability
  - Proper length and details
- Related work study

The structure of a research paper

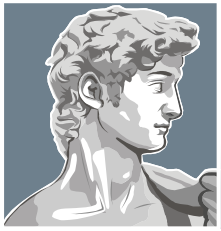
- Introduction
- Related work
- Methodology
- Experiments
- Discussion
- Conclusion & Future work

Writing is an art

- Charming introduction
  - Interesting and clear goal
  - Substantial motivation and challenges
  - Significant contribution
- Hierarchical structure
- Well positioned related work
- Clear definitions
- Thoughtfully beautiful figures
- Thorough and persuasive experiments
- Clearly claimed conclusion



- Thoughtfully beautiful pictures
- novel
- Insight
- Deliver key messages
- Hierarchical structure
- Skip some details
- revise the paper



- Attractive appearance
- Original
- Soul
- Highlight Key features
- Step-by-step carving
- Ignore some pores
- Polish a sculpture

Introduction - The most difficult part to write

- Background
- Goal
- Motivation
  - Highlight the key insight
  - Benefit and application scenarios
  - Difficulties and challenges
- Brief your method
- Contributions
- Structure of the rest

Finding efficient driving directions has become a daily activity and been implemented as a key feature in many map services like Google and Bing Maps. A fast driving route saves not only the time of a driver but also energy consumption (as most gas is wasted in traffic jams). In practice, big cities with serious traffic problems usually have a large number of taxis traversing on road surfaces. For the sake of management and security, these taxis have already been embedded with a GPS sensor, which enables a taxi to report on its present location to a data center in a certain frequency. Thus, a large number of time-stamped GPS trajectories of taxis have been accumulated and are easy to obtain.

Background

Motivation

Intuitively, taxi drivers are experienced drivers who can usually find out the fastest route to send passengers to a destination based on their knowledge (we believe most taxi drivers are honest although a few of them might give passengers a roundabout trip). When selecting driving directions, besides the distance of a route, they also consider other factors, such as the time-variant traffic flows on road surfaces, traffic signals and direction changes contained in a route, as well as the probability of accidents. These factors can be learned by experienced drivers but are too subtle and difficult to incorporate into existing routing engines. Therefore, these historical taxi trajectories, which imply the intelligence of experienced drivers, provide us with a valuable resource to learn practically fast driving directions.

Insight

Challenges

However, we need to face the following three challenges when performing our method.

Intelligence Modeling: As a user can select any place as a source or destination, there would be no taxi trajectory exactly passing the query points. That is, we cannot answer user queries by directly mining trajectory patterns from the data. Therefore, how to model taxi drivers' intelligence that can answer a variety of queries is a challenge.

Data Sparseness and Coverage: We cannot guarantee there are sufficient taxis traversing on each road segment even if we have a large number of taxis. That is, we cannot accurately estimate the speed pattern of each road segment.

Low-sampling-rate Problem: To save energy and communication loads, taxis usually report on their locations in a very low frequency, like 2-5 minutes per point. This increases the uncertainty of the routes traversed by a taxi[11]. As shown in Figure 2, there could exist four possible routes (R_1-R_4) traversing the sampling points a and b.

In this paper, we propose to mine smart driving directions from a large number of real-world historical GPS trajectories of taxis. As shown in Figure 1, taxi trajectories are aggregated and mined in the *Cloud* to answer queries from ordinary drivers or Internet users. Given a start point and destination, our method can suggest the practically fastest route to a user according to his/her departure time and based on the intelligence mined from the historical taxi trajectories. As the taxi trajectories are constantly updated in the *Cloud*, the suggested routes are state-of-the-art.

Goal

Introduction

- Contribution
  - How well it works
  - Support and validation

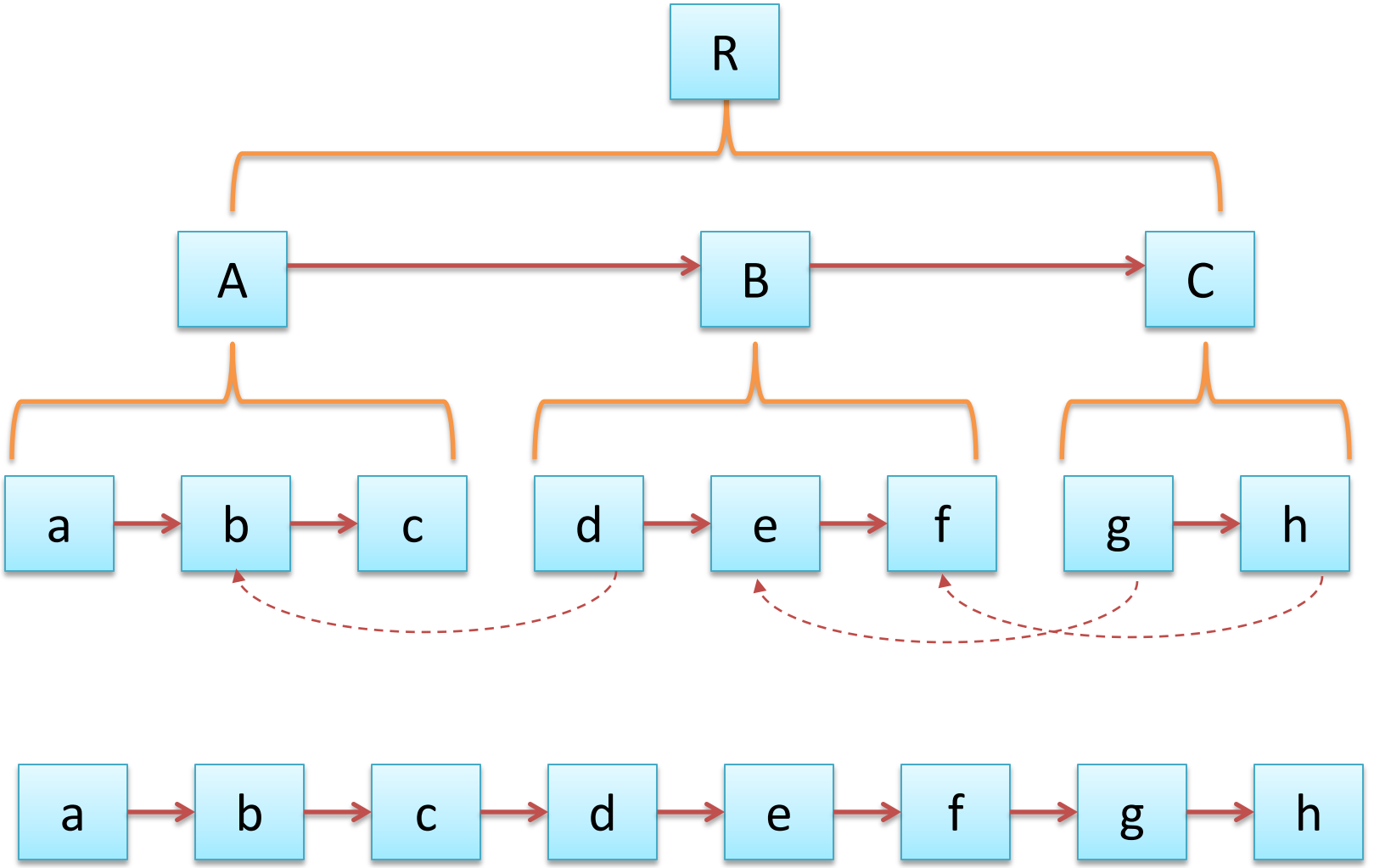
- We propose the notion of a landmark graph, which well models the intelligence of taxi drivers based on the taxi trajectories and reduces the online computation of route-finding.
- We devise Variance-Entropy-Based Clustering (called VE-Clustering) to learn the time-variant distributions of the travel times between any two landmarks.
- We build our system by using a real-world trajectory dataset generated by 33,000+ taxis in a period of 3 months, and evaluate the system by conducting both synthetic experiments and in-the-field evaluations (performed by real drivers). The results show that our method can find out faster routes with less online computation than competing methods.

Writing is an art

- Charming introduction
  - Interesting and clear goal
  - Substantial motivation and challenges
  - Significant contribution
- Hierarchical structure
- Well positioned related work
- Clear definitions
- Thoughtfully beautiful figures
- Thorough and persuasive experiments
- Clearly claimed conclusion


AUTOBOT IRONHIDE IS A GMC TOPKICK
AUTOBOT BUMBLEBEE IS A CHEVY CAMARO
AUTOBOT JAZZ IS A PONTIAC SOLSTICE
AUTOBOT RATCHET IS A HUMMER H2
TRANSFORMERS

Hierarchical Structure



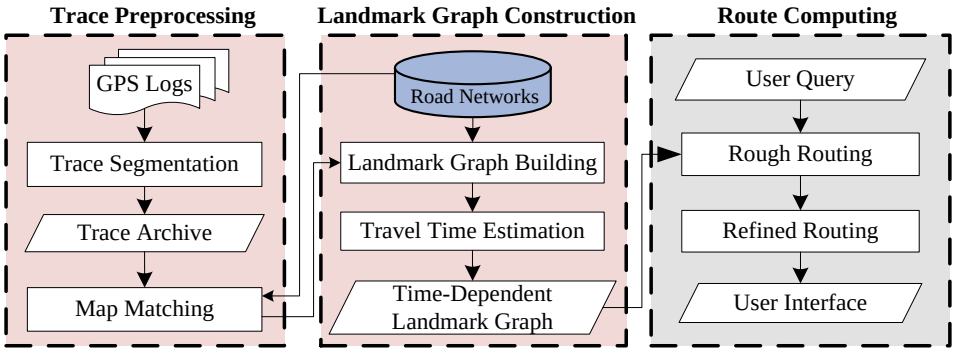
3. OVERVIEW

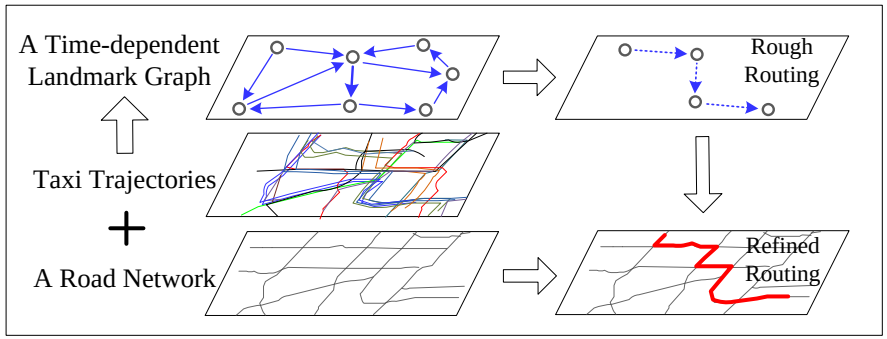
As shown in Figure 3, the architecture of our system consists of three major components: Trajectory Preprocessing, Landmark Graph Construction, and Route Computing. The first two components operate offline and the third is running online. The offline parts only need to be performed once unless the trajectory archive is updated.

Trajectory Preprocessing: This component first segments GPS trajectories into effective trips, then matches each trip against the road network. 1) *Trajectory segmentation*: In practice, a GPS log may record a taxi's movement of several days, in which the taxi could send multiple passengers to a variety of destinations. Therefore, we partition a GPS log into some taxi trajectories representing individual trips according to the taximeter's transaction records. There is a tag associated with a taxi's reporting when the taximeter is turn on or off, i.e., a passenger gets in or out of the taxi. 2) *Map matching*: We employ our IVMM algorithm [14], which has a better performance than existing map-matching algorithms when dealing with the low-sampling-rate trajectories, to map each GPS point of a trip to the corresponding road segment where the point was recorded. As a result, a taxi trajectory is converted to a sequence of road segments.

Landmark Graph Construction: We separate the weekday trajectories from the weekend ones, and build a *landmark graph* for weekdays and weekends respectively. When building the graph, we first select the top-k road segments with relatively more projections (i.e., being frequently traversed by taxis) as the *landmarks*. Then, we connect two landmarks with a *landmark edge* if there are a certain number of trajectories passing these two landmarks. Later, we estimate the distribution of travel time of each landmark edge by using the VE-clustering algorithm. Now, a time-dependent landmark graph is ready for online computation. Figure 4 demonstrates the key concept of our work.

Route Computing: Given a query (q_s, q_d, t_d), we carry out a two-stage routing algorithm to find out the fastest route. In the first stage, we perform a *rough routing* that





Writing is an art

- Charming introduction
  - Interesting and clear goal
  - Substantial motivation and challenges
  - Significant contribution
- Hierarchical structure
- Well positioned related work
- Clear definitions
- Thoughtfully beautiful figures
- Thorough and persuasive experiments
- Clearly claimed conclusion

Related work

- Categorize literatures
- Differentiate your work
  - Emphasize contributions
  - Justify the novelty
  - Position your work
- Position in a paper
  - Right after introduction
  - Right before conclusion

6. RELATED WORK

6.1 Mining Location History

***Mining individual location history*:** During the past years, a branch of research [5][9][11][13] has been performed based on individual location history represented by GPS trajectories. These works include detecting significant locations of a user [5][9], predicting the user's movement among these locations [5][13], and recognizing user-specific activities at each location [15]. As opposed to these works, we aim to model multiple users' location histories and learn patterns from numerous individuals' behaviors.

***Mining multiple users' location histories*:** Gonotti et al. [8] mined similar sequences from users' moving trajectories, and Mamoulis et al. [16] proposed a framework for retrieving maximum periodic patterns in spatio-temporal data. MSMLS [11] used a history of a driver's destinations, along with data about driving behavior extracted from multiple users' GPS trajectories, to predict where a driver may be going as a trip progresses. Eagle et al [7] aimed to recognize the social pattern in daily user activity from the dataset collected by 100 users with a Bluetooth-enabled mobile phone. In contrast to these techniques, we extend the paradigm of mining multiple users' location histories from exploring users' behaviors to understanding locations as well as modeling the relationship between users and locations.

6.2 Location Recommenders

***Recommenders based on real-time location*:** Mobile tourist guide systems [4][6][14][17] typically recommend locations and sometimes provide navigation information based on a user's real-time location. Previously, such kinds of systems were somehow naïve as they always returned the information close to an individual without understanding the individual and the nearby locations. Recently, some researchers aim to filter away from the returned results the invisible entities occluded by the nearby building [6][17]. Meanwhile, another branch of work [4][14] started involving a user's location history in these systems to provide the user with a more personalized recommendation. In contrast to these techniques, we aim to integrate social networking into the mobile tourist guide systems, by helping each individual deeply understand the locations around them with the knowledge mined from multiple users' location histories.

***Recommenders based on location history*:** Using multiple users' real-world location histories, some recommender systems, such as *Geowhiz* [10] and *CityVoyager* [18], etc, have been designed to recommend geographic locations like shops or restaurants to users. Horozov et al. [10] proposed an enhanced collaborative filtering solution to generate the recommendation of a restaurant. Takeuchi et al. [18] attempted to recommend shops to users based on their individual preferences estimated by analyzing their past location histories. The major difference between these works and ours lies in two aspects. First, we differentiate the travel experiences of various users. Second, we consider the relationship between locations and users' travel experiences, e.g., the mutual reinforcement relationship and the region-related constraints.

Writing is an art

- Charming introduction
  - Interesting and clear goal
  - Substantial motivation and challenges
  - Significant contribution
- Hierarchical structure
- Well positioned related work
- Clear definitions
- Thoughtfully beautiful figures
- Thorough and persuasive experiments
- Clearly claimed conclusion

Clear definition

- Define it until you need it
- Use figures to demonstrate

Definition 1: *Road Segment* (r). A road segment is a directed edge that is associated with a direction symbol $(r.dir$, one-way or two-way), two terminal points $(r.s, r.e)$, and a list of intermediate points describing the segment using a polyline. Additional information is maintained for each road segment: level $(r.level)$ and length $(r.length)$, e.g., a high way's level is usually 0 and that of a ring road in a city is 1.

Definition 2: *Road Network* (G_r). A road network G_r is a directed graph, $G_r = (V_r, E_r)$, where E_r is the set of edges representing road segments, and V_r is the collection of terminal points of corresponding road segments.

Definition 3: *Taxi Trajectory* $(Traj)$. A taxi trajectory is denoted as a sequence of GPS points, i.e., $Traj: p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$. Each point is represented as $p_i = (l, t, v, s)$, where l denotes the location of a taxi at time t, travelling with (instantaneous) speed of v, given state s. Note that the state of taxi s can be occupied (O), or not-occupied (N), or parked (P).

Definition 4: *Parking Place* (Pk). Given a trajectory, $Traj: p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, a parking place is a sub-trajectory $Traj: p_i \rightarrow \cdots \rightarrow p_j$, which satisfies the conditions where $\forall k \in [i, j), distance(p_k, p_{k+1}) < \delta$, $t_j - t_i > \tau$ and $speed(p_i) < \varepsilon$.

Definition 5: *Taxi Segment*. A taxi segment is the sub-trajectory between two parking places. A taxi segment could contain one or more trips, each of which is comprised of the GPS points with the same state. Refer to Fig. 2 for an example.

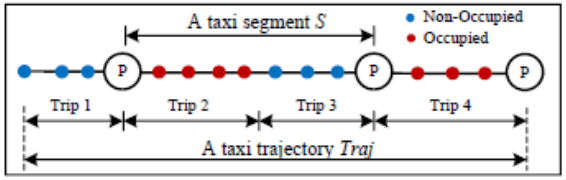


Fig. 2. Notations used in this paper

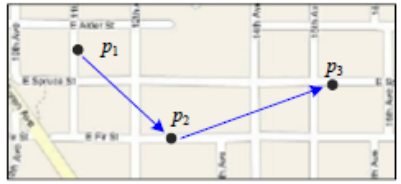


Fig. 3 A low-sampling-rate trajectory

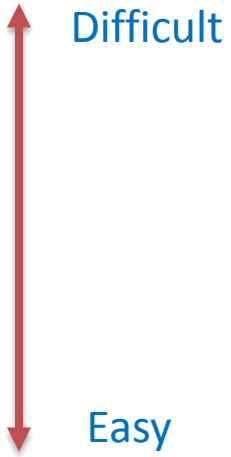
Problem Definition: Given a taxi trajectory $Traj$, we first detect from $Traj$ some parking places (P) dividing $Traj$ into taxi segments. Then, we infer the state, non-occupied (N) or occupied (O) by passengers, of each point from each taxi segment given a road network G_r, a POI dataset, and a set of historical taxi trajectories with state labels.

Writing is an art

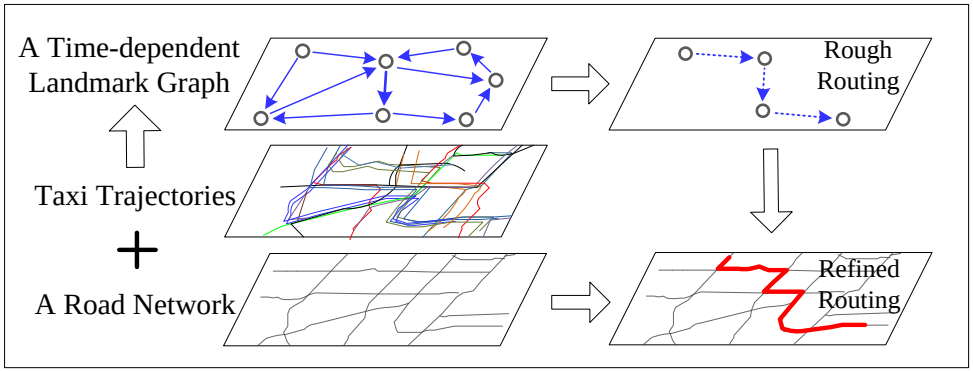
- Charming introduction
  - Interesting and clear goal
  - Substantial motivation and challenges
  - Significant contribution
- Hierarchical structure
- Well positioned related work
- Clear definitions
- Thoughtfully beautiful figures
- Thorough and persuasive experiments
- Clearly claimed conclusion

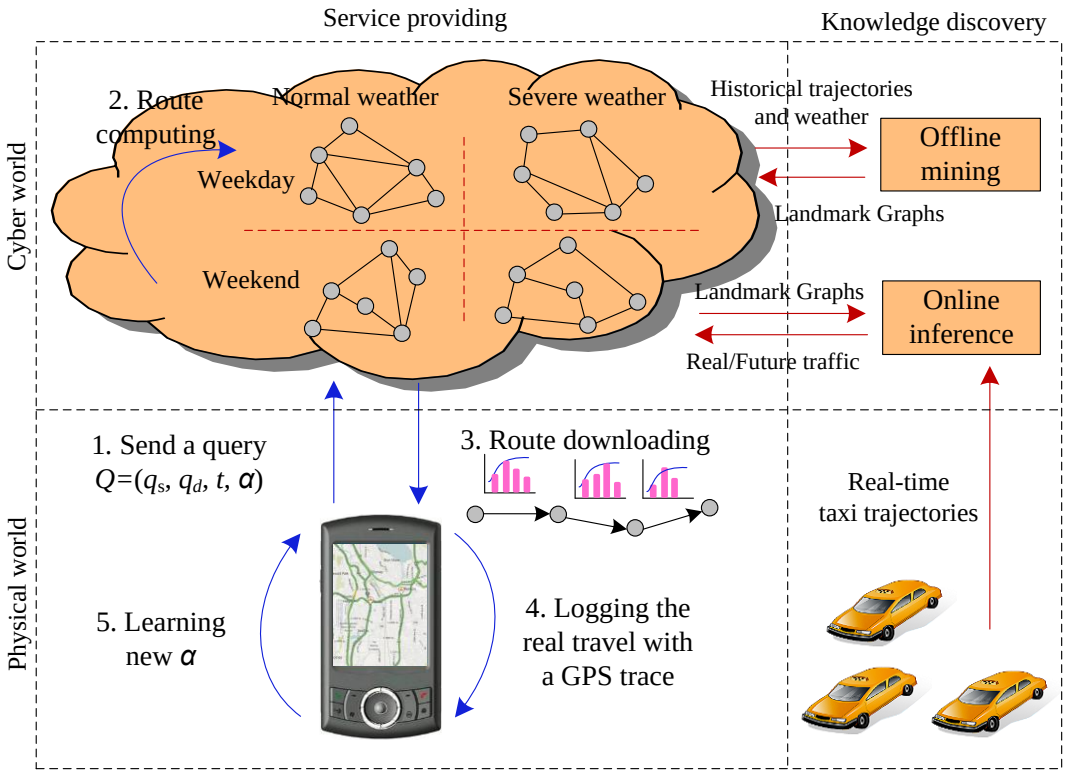
Thoughtfully beautiful figures

- Substantiate vision and concept
- Presenting framework and architecture
- Illustrating methodology
- Showing results

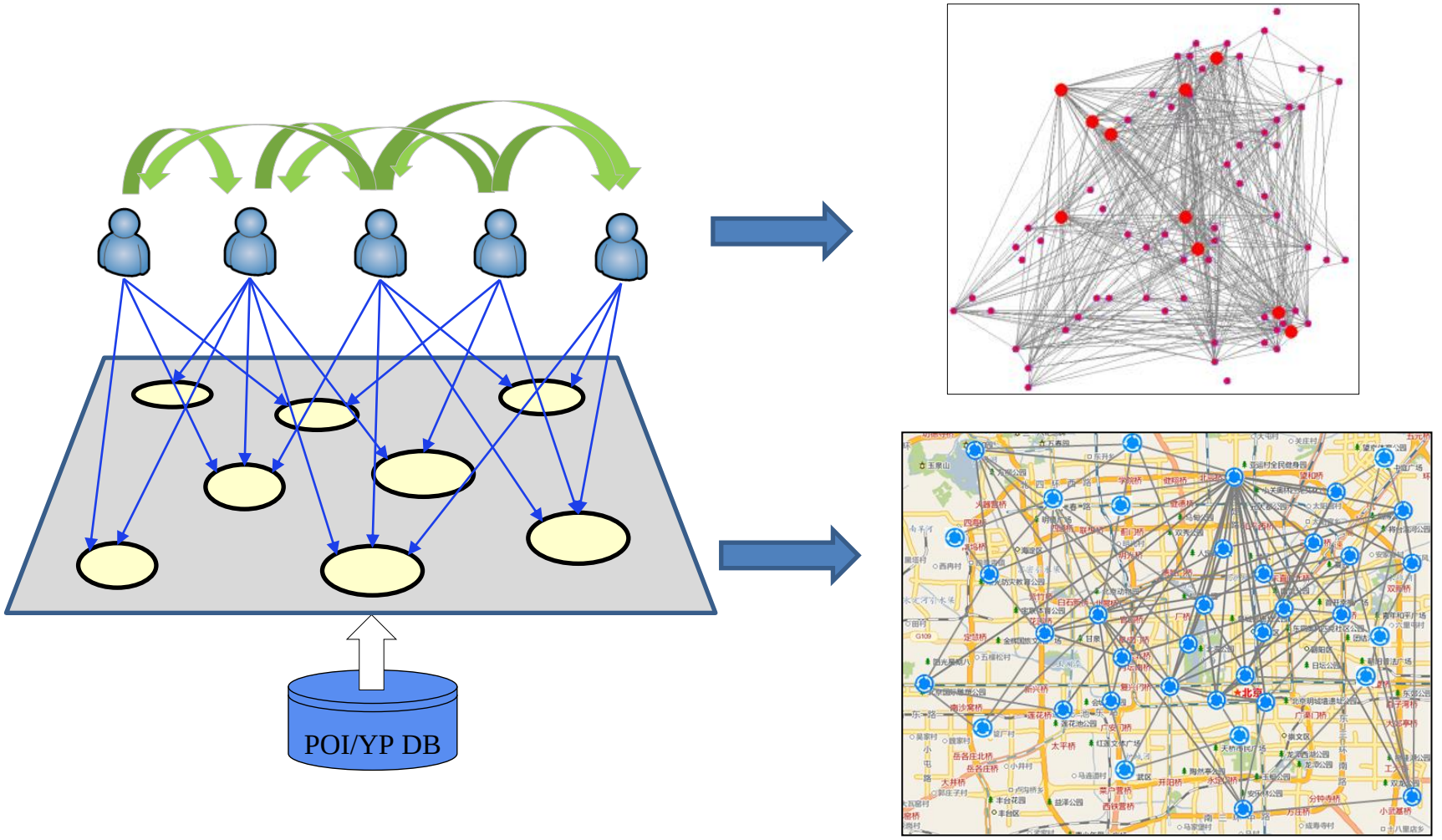


Substantiate idea and concept



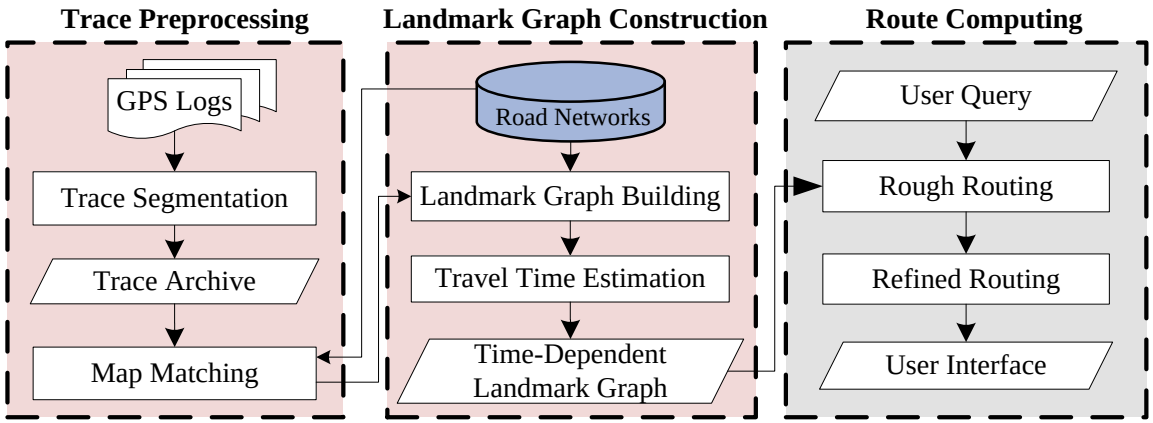


GeoLife: A Location-History-Based Social Network

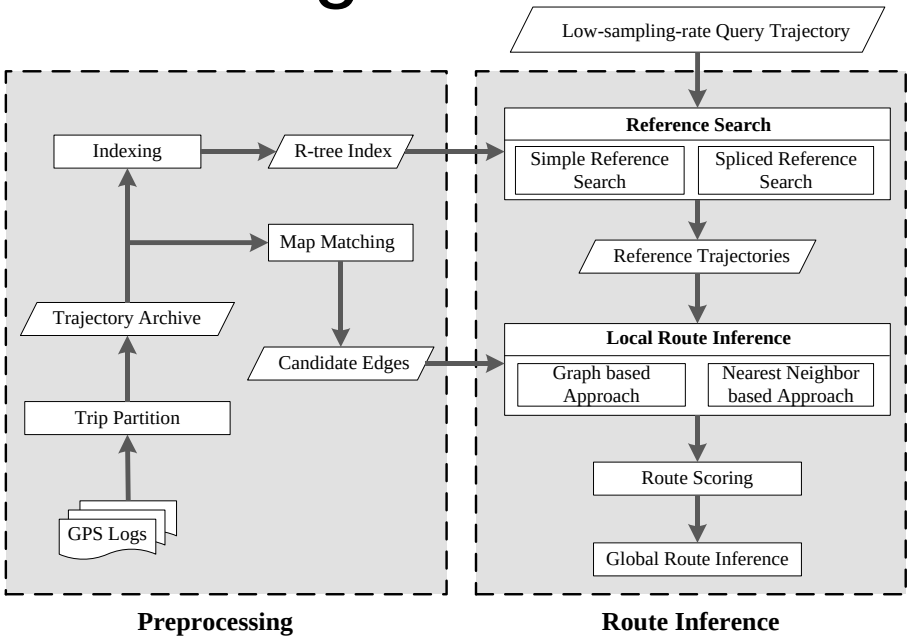


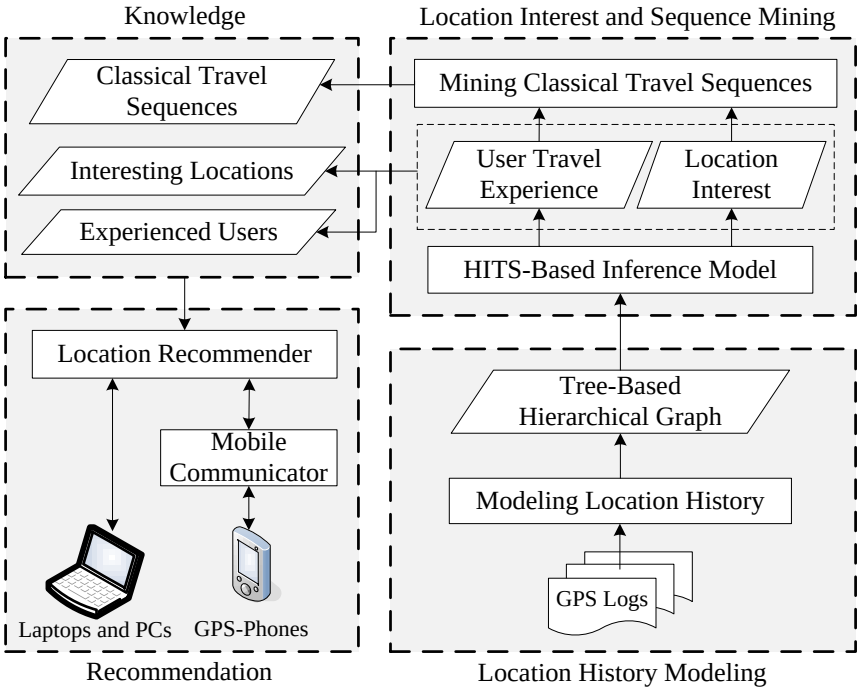
Presenting framework

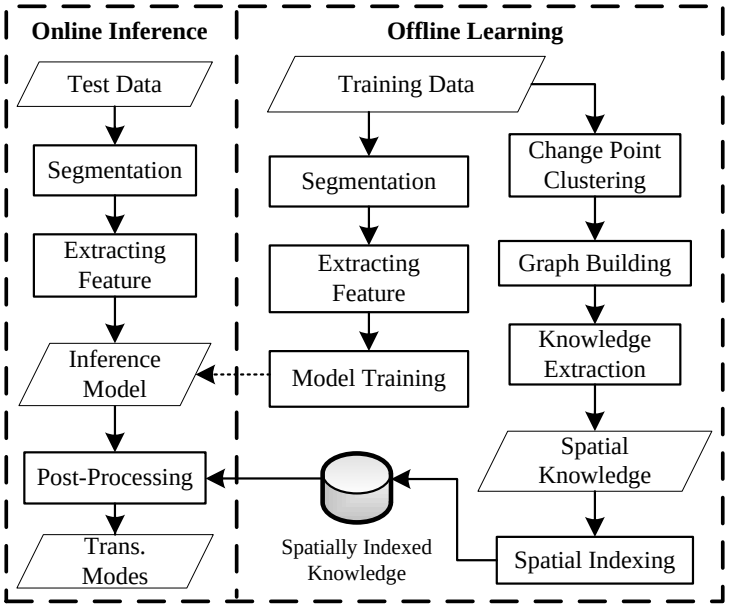
- Modularization
- Mapping to paper sections

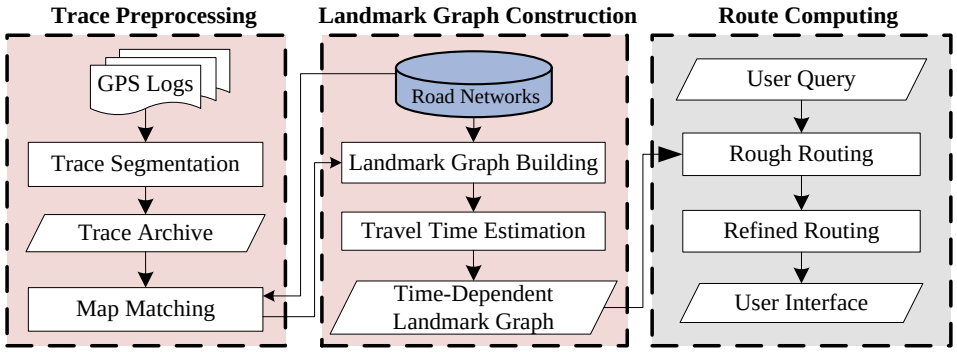


Presenting framework

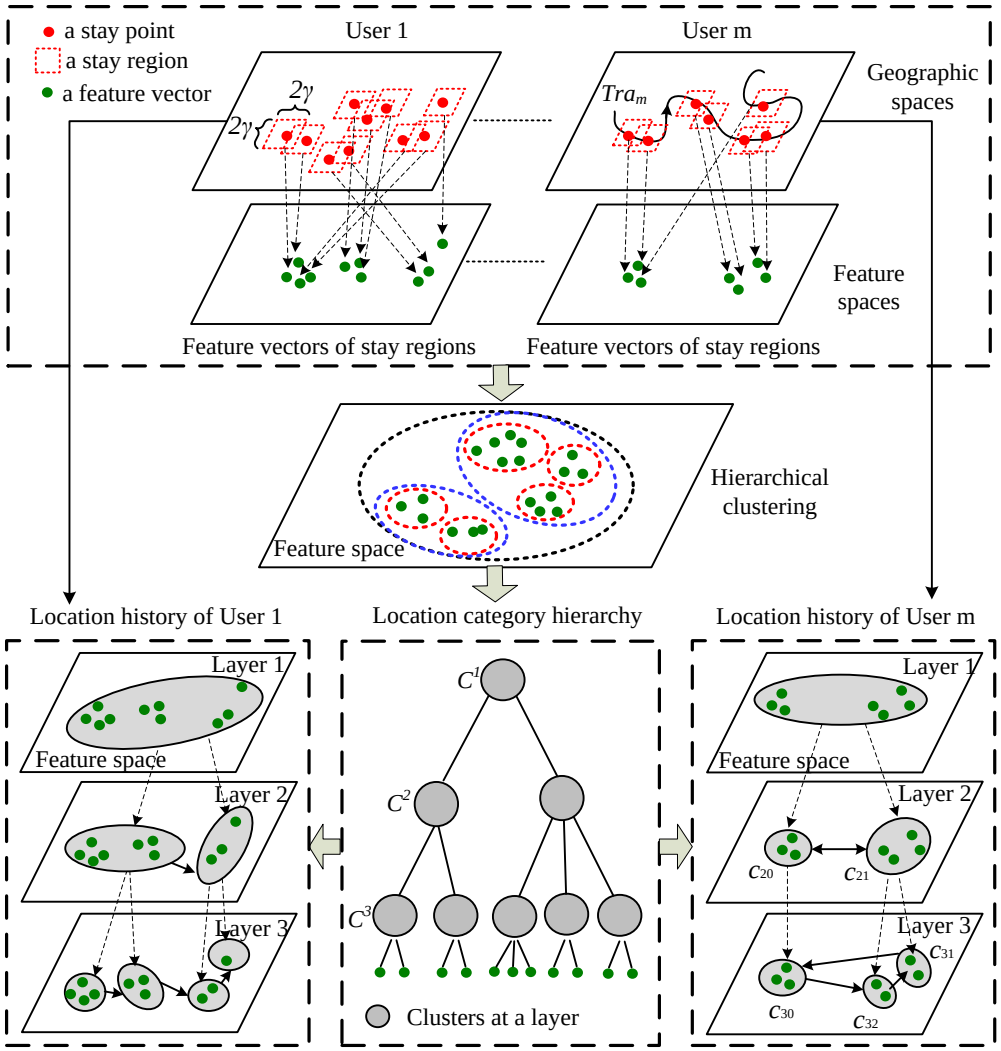






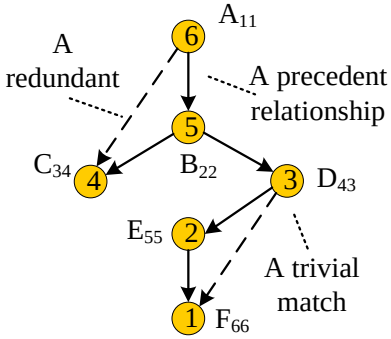


Illustrating methodology

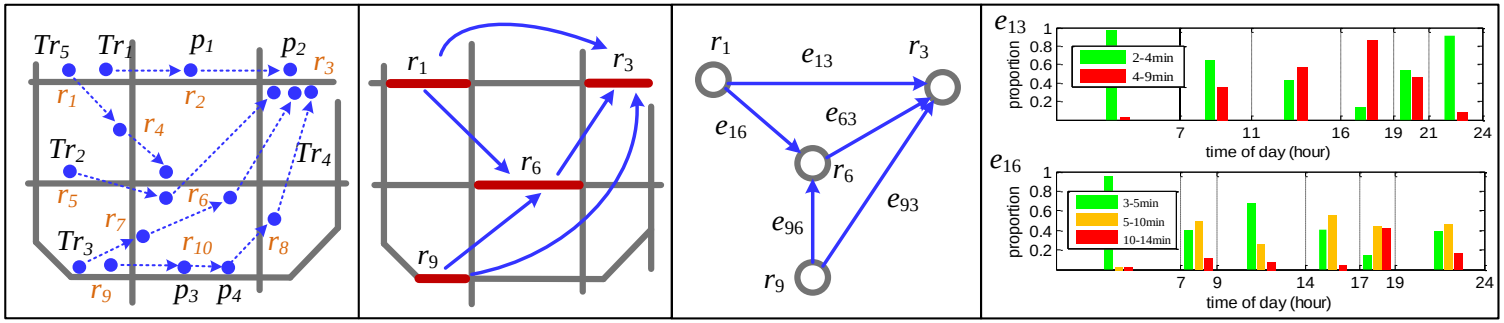


Index		1 A	2 B	3 C	4 D	5 E	6 F
1	A	1	0	0	0	0	0
2	B	0	1	0	0	0	0
3	D	0	0	0	1	0	0
4	C	0	0	1	0	0	0
5	E	0	0	0	0	1	0
6	F	0	0	0	0	0	1
7	G	0	0	0	0	0	0

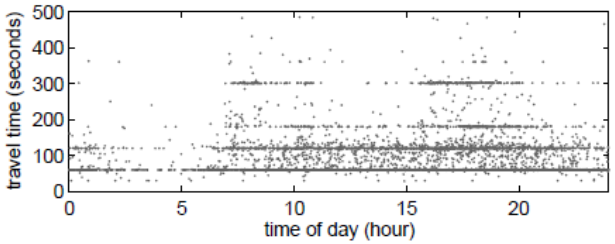
(a) The match matrix



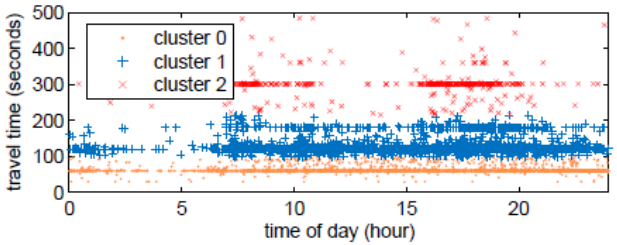
(b) The precedent graph



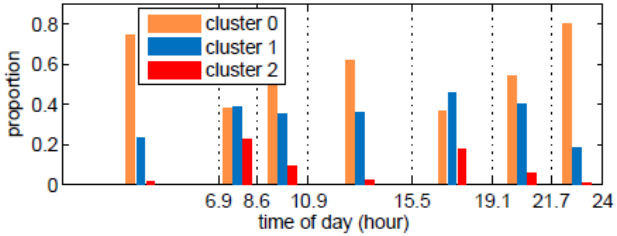
A) Matched taxi trajectories B) Detected landmarks C) A landmark graph D) Travel time estimation



(a) Transitions of a landmark Edge



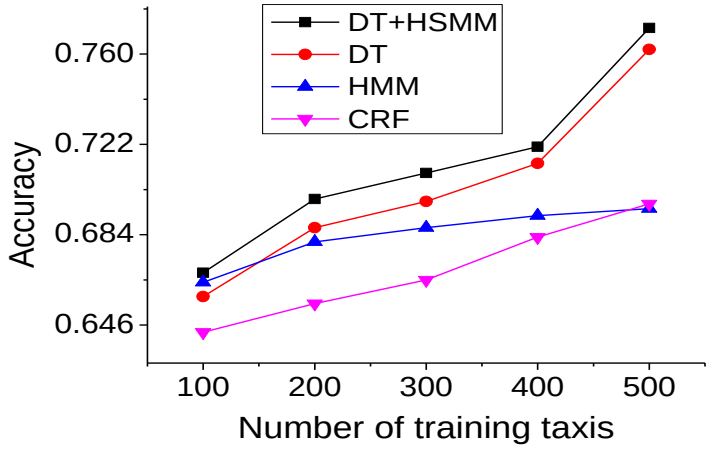
(b) V-Clustering result

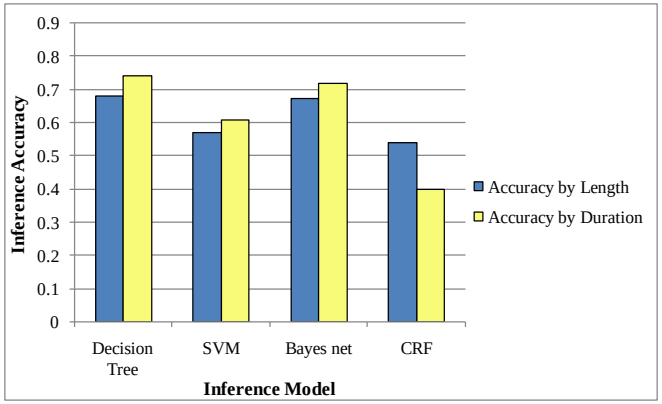


(c) VE-Clustering result

Figures showing results

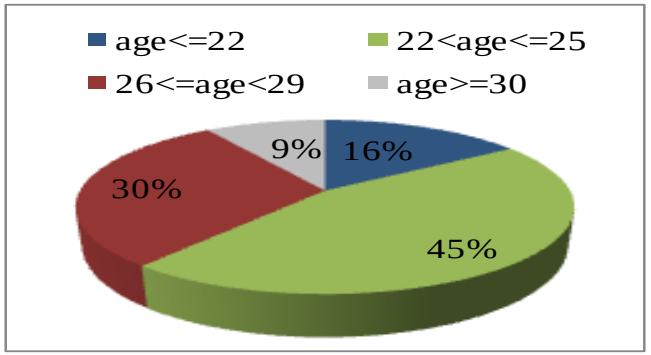
- Curve
- Bar
- Pie
- Table
- Visualization





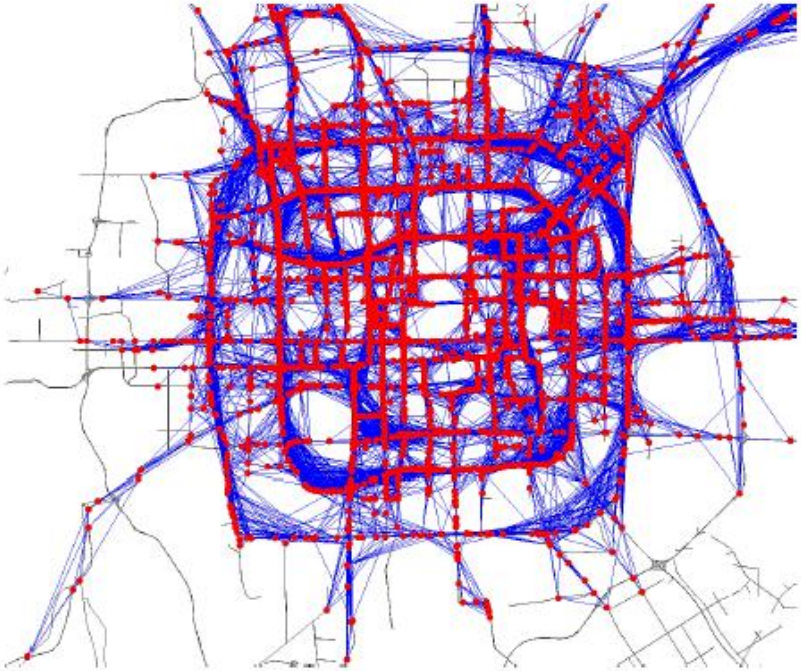
Ground truth	Predicted Results (KM)				
	Walk	*Driving*	*Bus*	*Bike*	
Walk	1026.4	122.1	386.5	357.3	0.543
Driving	42.6	2477.3	458.5	235.1	0.771
Bus	34.8	164.7	1752.4	46.2	0.877
Bike	49.3	113.5	31.9	1234.3	0.864
	0.891	0.861	0.666	0.659	**0.762**
	Precision				

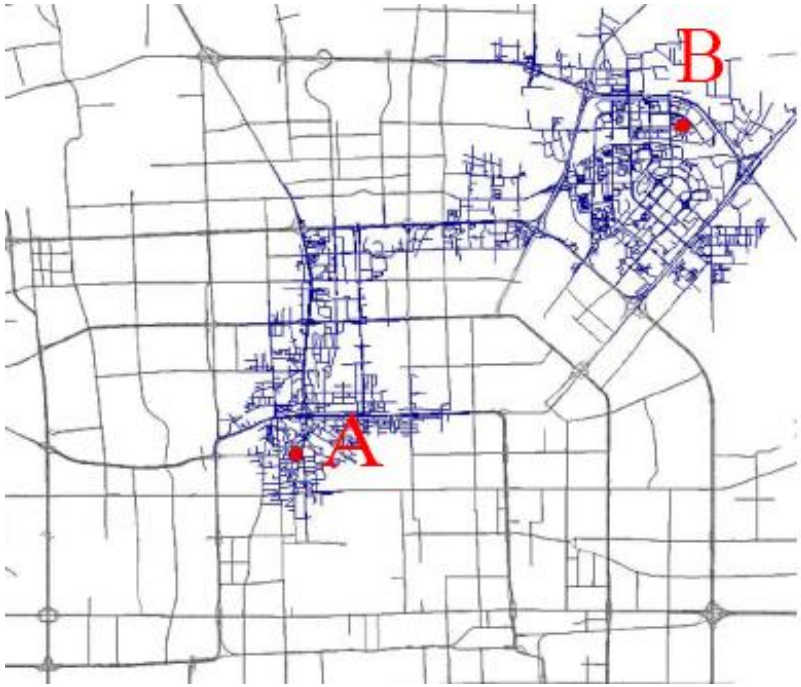
Recall



Figures showing results

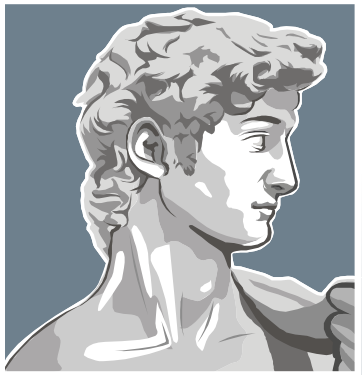
- Visualization





Writing is an art

- Charming introduction
  - Interesting and clear goal
  - Substantial motivation and challenges
  - Significant contribution
- Hierarchical structure
- Well positioned related work
- Clear definitions
- Thoughtfully beautiful figures
- Thorough and persuasive experiments
- Clearly claimed conclusion



Thorough and persuasive evaluation

- Settings
  - Datasets
  - Parameters
  - Users and objects
- Approach
  - Strategy
  - Metrics
  - Ground truth
  - Baselines
- Results
  - Data → conclusion
  - Analysis and reasons
  - Compare with baselines
  - Self-exploration

Writing is an art

- Charming introduction
  - Interesting and clear goal
  - Substantial motivation and challenges
  - Significant contribution
- Hierarchical structure
- Well positioned related work
- Clear definitions
- Thoughtfully beautiful figures
- Thorough and persuasive experiments
- Clearly claimed conclusion

Clearly claimed conclusion

- Emphasize contribution and novelty
- Highlight key results with numbers
- Be careful using auxiliary verb
  - Weak: can, may, might, would, could…
  - Strong: did, do, will

This paper presents an approach that finds out the practically fastest route to a destination at a given departure time in terms of taxi drivers' intelligence learned from a large number of historical taxi trajectories. In our method, we first construct a time-dependent landmark graph, and then perform a two-stage routing algorithm based on this graph to find the fastest route. We build a real system with real-world GPS trajectories generated by over 33,000 taxis in a period of 3 months, and evaluate the system with extensive experiments and in-the-field evaluations. The results show that our method significantly outperforms both the speed-constraint-based and the real-time-traffic-based method in the aspects of effectiveness and efficiency. Given over 5 taxis in a region of $1km^2$, more than 60% of our routes are faster than that of the speed-constraint-based approach, and 50% of these routes are at least 20% faster than the latter. On average, our method can save about 16% of time for a trip,

How to present a component

- Step-wise
- WWHW
  - What to do
  - Why
  - How to
  - Why
- Using figures and running examples

In this step, we first detect parking place candidates by using the density-based clustering algorithm demonstrated in Fig. 6 A-D) and formally described in Fig. 7 A). Then we remove some false candidates caused by traffic jams or traffic lights from the candidate set according to the filtering algorithm depicted in Fig. 6 E-F) and defined in Fig. 7 B).

Fig. 6 demonstrates the parking place candidate detection algorithm, using a taxi trajectory ($p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_7$) as an example. First, we keep on checking the speed of each point and the distance between this point and the point next to it until both values (speed and distance) are smaller than the corresponding thresholds. As depicted in Fig. 6 B), p_1 and p_2 cannot formulate a parking place as the distance between them $dist(p_1, p_2)$ exceeds the corresponding threshold δ. After that, we move to p_2 and find that $dist(p_2, p_3) < \delta$, $dist(p_2, p_4) < \delta$ while $dist(p_2, p_5) > \delta$ (Fig. 6 C)). If the speed values of these three points are less than the threshold ϵ and the time interval between $p_2.t$ and $p_4.t$ is larger than the time threshold τ, the three points form a small cluster representing a parking place candidate. However, they might not be the entire set of the points in this parking place. Therefore, we keep on expanding the parking place by continuously checking the distance between p_3 and the remaining points in the trajectory (p_4, p_5, p_6, p_7). As depicted in Fig. 6 D), p_5 and p_6 are added into the parking set since they also meet the speed, distance and time interval constraints. Finally, we detect $(p_2, p_3, p_4, p_5, p_6)$ as a parking place candidate because we cannot expand the cluster any further, i.e., all the points in the cluster have a distance farther than δ to p_7.

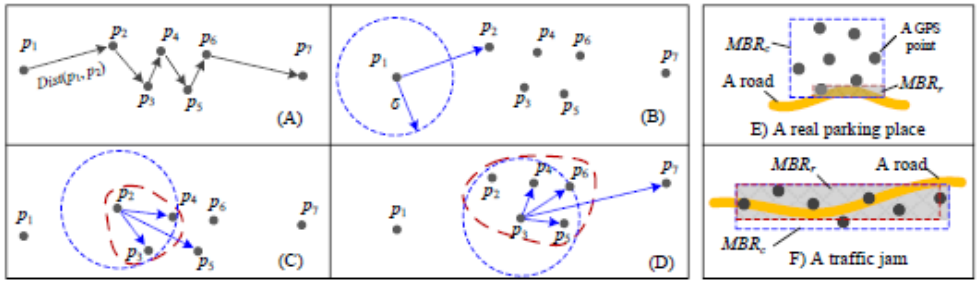


Fig. 6. A demonstration of parking place detection

Essentially, the detection algorithm finds out the locations where the GPS points of a taxi are densely clustered, with spatial, temporal and speed constraints. This algorithm can find out both the places where a taxi remains stationary, such as a taxi stand, and the queue-structured regions in which a taxi keeps on moving forward slowly, like the taxi queue in an airport.

However, a parking place candidate could sometimes be generated by taxis stuck in a traffic jams, or waiting for signals at a traffic light, instead of a real parking (as defined in Definition 4). To reduce such false positives, we propose a filtering algorithm that differentiates the real parking places close to a street from the above-mentioned scenarios, as illustrated in Fig. 6 E) and F). Intuitively, the minimal bounding box (MBR_c) of the GPS points generated in a real parking place close to a

What: detect parking places

Why: segment trajectories and reduce complexity

Why using the method: traffic jams and K-mean does not work

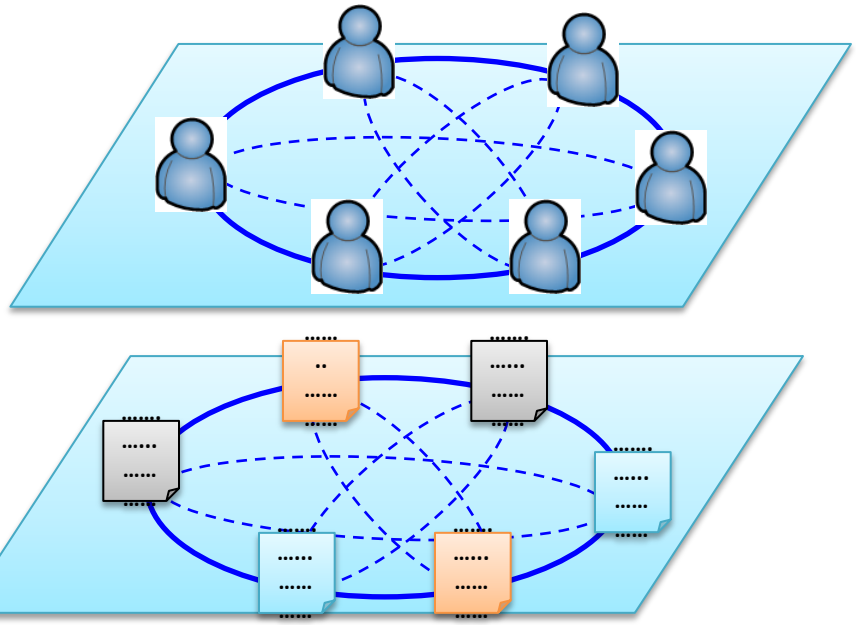
How to

Warnings

- Many details: A research paper is not a specification
- Missing key messages
- Presenting a simple problem in a complex way
- Use too many symbols
- Criticize other people's work
- Over claim and excessive contribution

Take away

- People
- Structure
- Figures
- Related work



Thanks!

Yu Zheng, Web Search & Mining Group

http://research.microsoft.com/en-us/people/yuzheng/