Destination Prediction by Sub-Trajectory Synthesis and Privacy Protection Against Such Prediction

Andy Yuan Xue [#1], Rui Zhang [#2], Yu Zheng [‡3], Xing Xie [‡4],
Jin Huang [#5], Zhenghua Xu [#6]

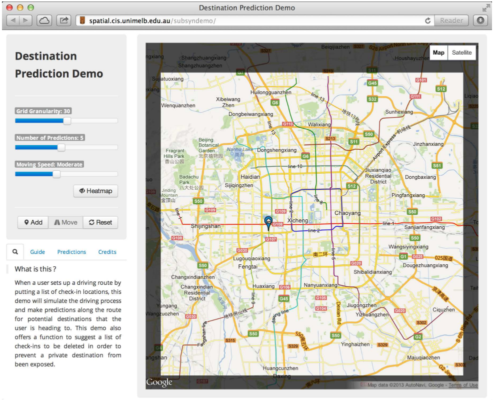
[#] *University of Melbourne, Victoria, Australia*
[1] andy.xue@unimelb.edu.au [2] rui@csse.unimelb.edu.au
[5] jin.h@iojin.com [6] zhxu@student.unimelb.edu.au

[‡] *Microsoft Research Asia, Beijing, P.R.China*
[3,4] {yuzheng, xingx}@microsoft.com

Introduction

Purpose: To predict destinations of travel based on public data.

A demo: Visitor drives from the Forbidden Palace in Beijing to the International Airport.



Introduction

Applications:

- Recommend sightseeing places
- Send targeted advertisements
- Automatically set destinations and route in navigation systems

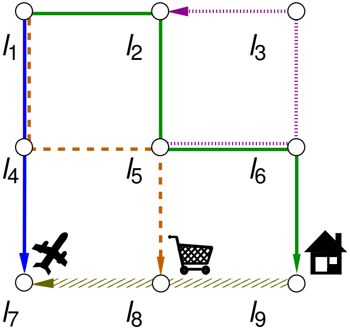
Introduction

An example of a baseline solution adapted from existing work:

- Grid representation
- Trajectory matching
- A user travels from l_1 to l_4: Predicted destinations l_7 and l_8
- Query trajectory $\{l_1, l_2, l_3\}$: no predicted destination due to lack of training data.
- Baye's rule

$$P(d \in l_j | T^p) = \frac{P(T^p | d \in l_j) \cdot P(d \in l_j)}{\sum_{k=1}^{g^2} P(T^p | d \in l_k) \cdot P(d \in l_k)}$$

- **Data Sparsity Problem**



Destination Prediction

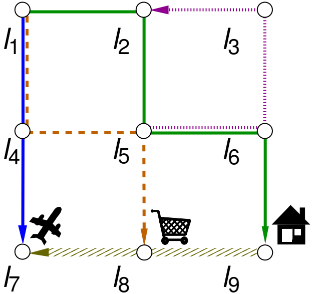
Sub-Trajectory Synthesis (SubSyn):

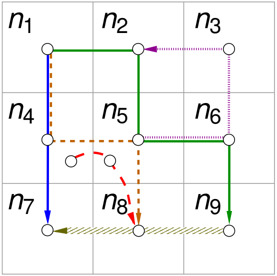
- Solves the data sparsity problem by expanding the historical dataset.
- Two phases: Decomposition and Synthesis

Destination Prediction

Sub-Trajectory Synthesis (SubSyn): **Decomposition**

- Partition and group POIs into grid cells.

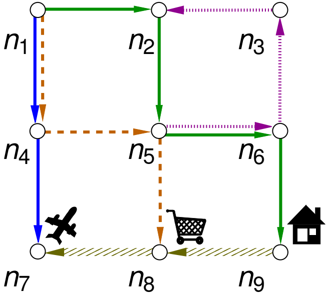


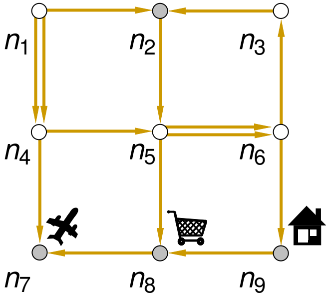


Destination Prediction

Sub-Trajectory Synthesis (SubSyn): **Decomposition**

- Partition and group POIs into grid cells.
- Decompose historical trajectories into sub-trajectories.

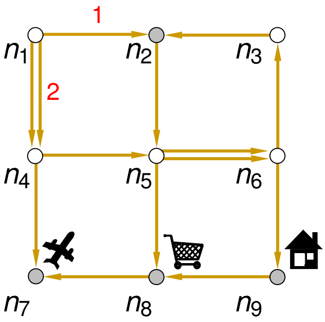




Destination Prediction

Sub-Trajectory Synthesis (SubSyn): **Decomposition**

- Use Markov model
- Transition matrix M: p_{12}, p_{14}, p_{78}, etc.



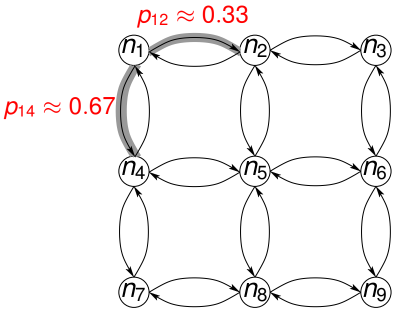
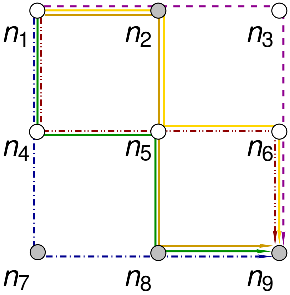


Figure: 3×3 Markov model

Destination Prediction

Sub-Trajectory Synthesis (SubSyn): **Synthesis**

- Starting from n_1, what is the probability of travelling to n_9?
- Shortest Path is 4: $p_{1\to 9} = M^4_{1,9}$
- M^4: transition between cells with distance 4.



- Consider detour (within 1.2 times shortest path. $\alpha = 0.2$)
- Users may travel either distance 4 or 5 ($\lceil 4 \times 1.2 \rceil$) to reach n_9: $p_{1\to 9} = M^4_{1,9} + M^5_{1,9}$

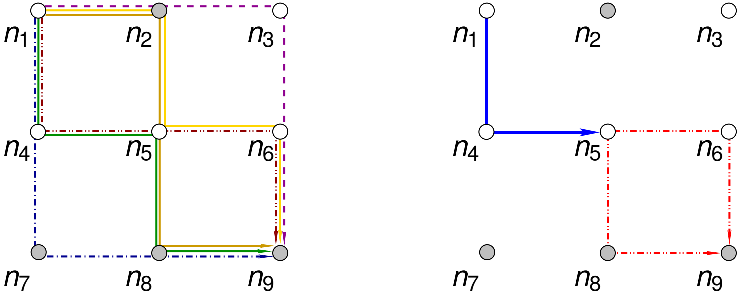
Destination Prediction

Sub-Trajectory Synthesis (SubSyn): **Synthesis**

- Given a user's route: $T^p = \{n_1, n_4, n_5\}$,
- The probability of n_9:

$$P(n_9|T^p) = P(n_9|n_1, n_4, n_5)$$
$$\propto \frac{p_{5\to9}}{p_{1\to9}} \cdot P(n_9|n_1)$$

(derivation in paper using Bayes' rule)



Algorithms

$$P(n_k|T^p) \propto \frac{p_{c\to k}}{p_{s\to k}} \cdot P(n_k|n_s)$$

- Two stages: Training and Prediction
- SubSyn-Training constructs Markov model and computes various probabilities needed for prediction. (RHS of the equation)
- Efficiently perform huge matrix multiplications. E.g., compute M^{100} where M is a 2500×2500 matrix.
- SubSyn-Prediction retrieves these probabilities to compute the destination probabilities $P(n_k)|T^p)$

Privacy Protection
Demo

A demo: check-ins on your way home.

Privacy Protection
Methods

Exhaustive Generation Method

- Iteratively delete each node in query trajectory
- Inefficient

End-Points Generation Method

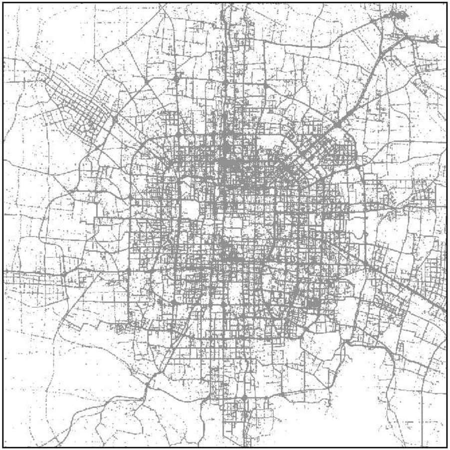
- **Theorem:** Only the starting and current nodes affect the probabilities of predicted destinations
- Is a property of first-order Markov model
- Dramatically reduced search space, efficient for online queries

Experimental Study
Dataset

Real-world taxi trajectory dataset in the city of Beijing.

Contains:

- 580,000 taxi trajectories
- 5 million kilometres of distance travelled



Experimental Study

Grid Granularity

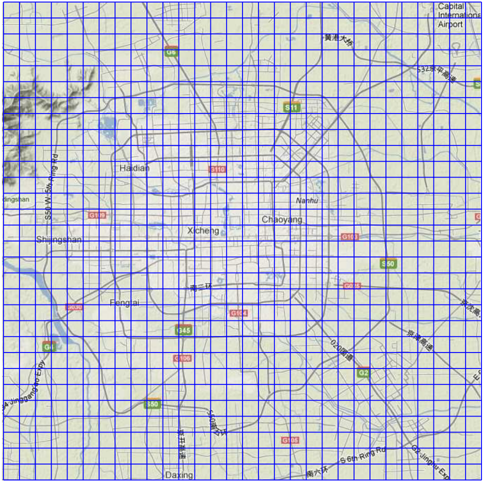
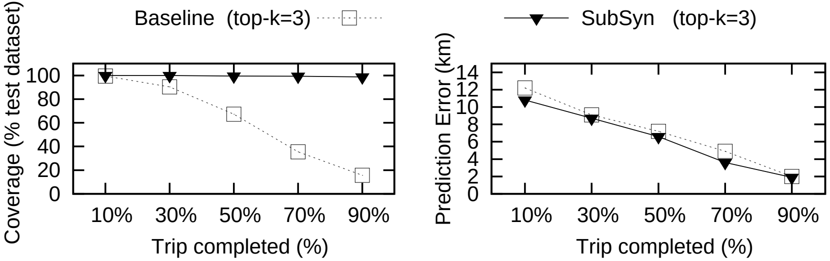


Figure: Map of Beijing with 30 $\times$ 30 grid overlay: Each cell $\approx 1.78km^2$

Experimental Study
Effectiveness

- Randomly pick 1000 test/query trajectories
- Algorithms: Existing vs SubSyn
- Measurements: Coverage and Prediction Error



More experiments in the paper

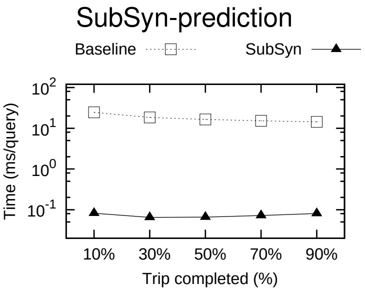
Experimental Study

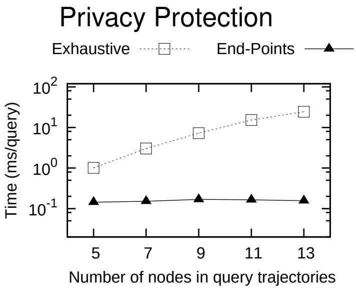
Runtime Efficiency

SubSyn-Training

Grid Granularity	20	**30**	40	50
Running Time (hours)	0.03	**0.5**	3	17

- Commodity computer: Intel i7-860 CPU 4GB RAM





Conclusion

- Identified Data Sparsity Problem, and proposed a Sub-Trajectory Synthesis (SubSyn) algorithm which successfully addressed the problem.
- SubSyn decomposes historical trajectories into sub-trajectories to exponentially increase practicality.
- SubSyn can predict destinations for up to ten times more query trajectories than the existing algorithm.
- Runs over two orders of magnitude faster constantly.
- Also proposed an efficient method (two orders of magnitude faster) to avoid privacy leak.

Questions

Questions?

Demo:
http://spatialanalytics.cis.unimelb.edu.au/subsyndemo/

Contacts:

Andy Yuan Xue andy.xue@unimelb.edu.au
http://people.eng.unimelb.edu.au/yuanx/

Rui Zhang rui.zhang@unimelb.edu.au
http://people.eng.unimelb.edu.au/zr/

References:

- Andy Yuan Xue, Rui Zhang, Yu Zheng, Xing Xie, Jin Huang, Zhenghua Xu. **Destination Prediction by Sub-Trajectory Synthesis and Privacy Protection Against Such Prediction**. IEEE International Conference on Data Engineering (**ICDE**) 2013.
- Andy Yuan Xue, Rui Zhang, Yu Zheng, Xing Xie, Jianhui Yu, Yong Tang. **DesTeller: A System for Destination Prediction Based on Trajectories with Privacy Protection**. International Conference on Very Large Data Bases (**VLDB**) 2013 (Demo)