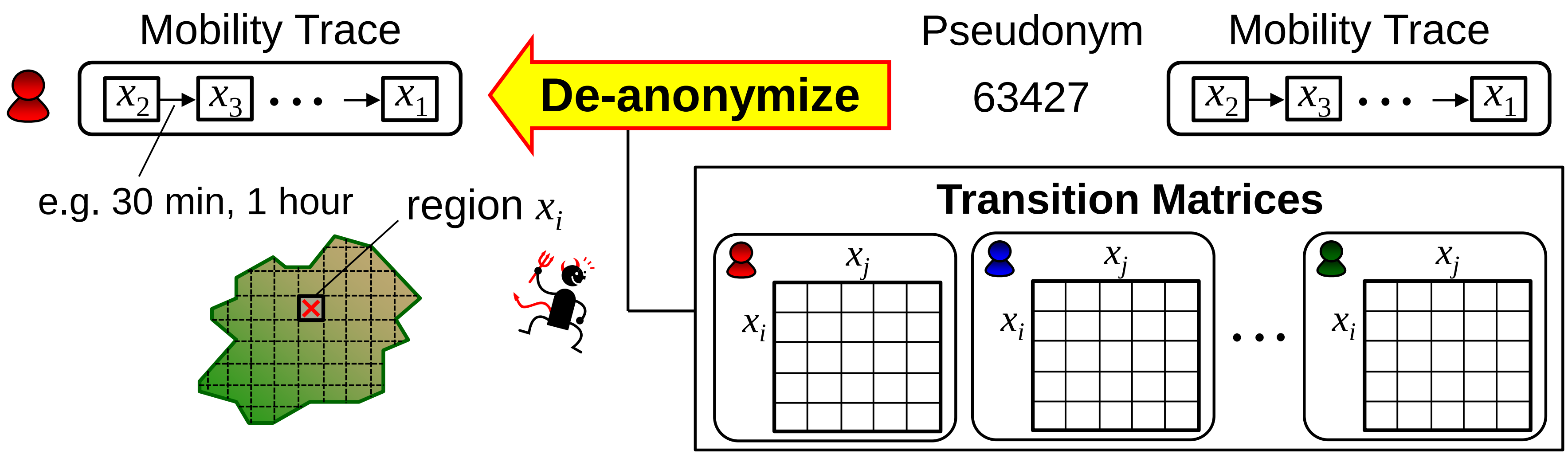
Expectation-Maximization Tensor Factorization for Practical Location Privacy Attacks

Takao Murakami (AIST*, Japan)

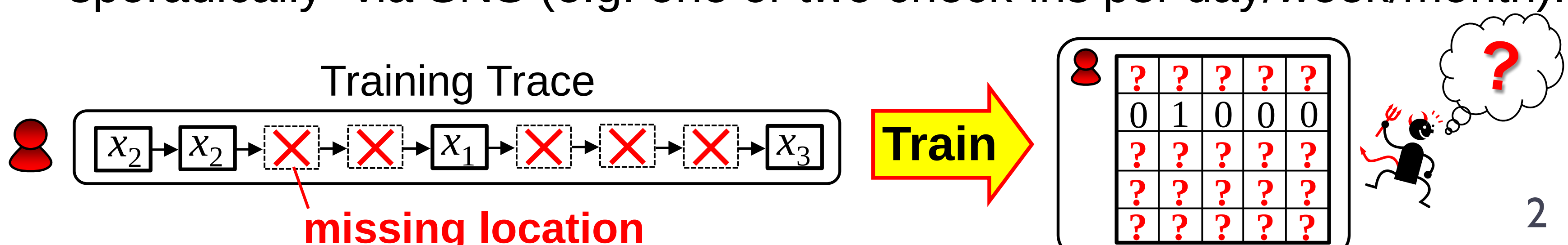
*AIST: National Institute of Advanced Industrial Science & Technology

Outline

- Markov Chain Model-based Attacks [Shokri+,S&P11] [Gambs+,JCSS14] [Mulder+,WPES08] [Xue+,ICDE13] etc.
  - Attacker can de-anonymize traces (or infer locations) with high accuracy when the amount of training data is very large.

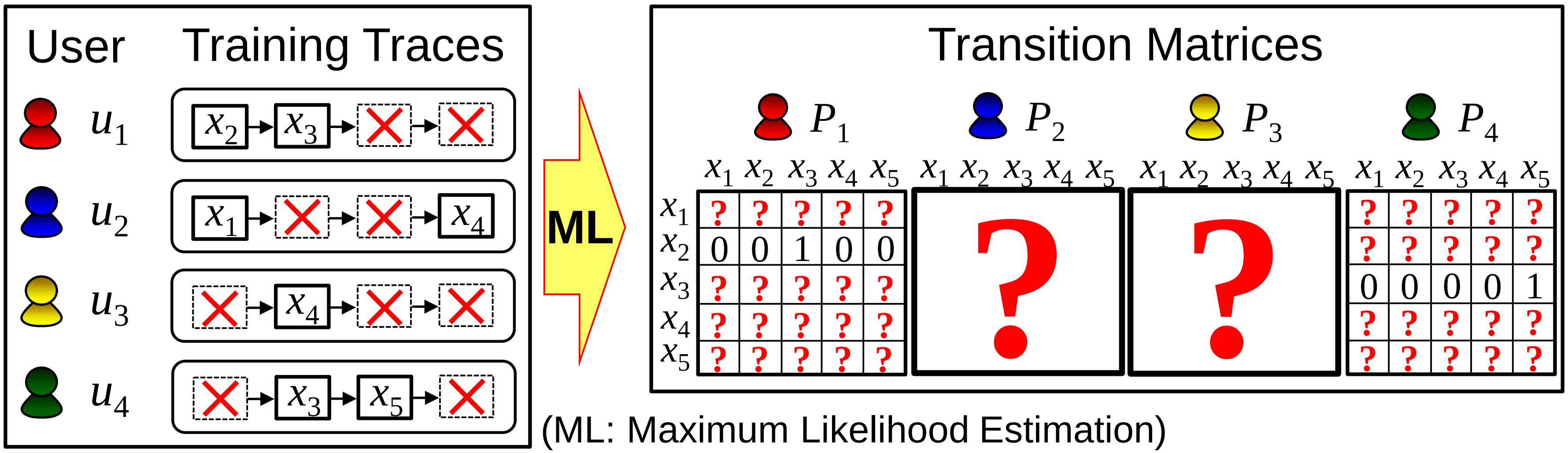


- In reality, training data can be sparsely distributed over time…
  - Many users disclose a small number of locations not continuously but "sporadically" via SNS (e.g. one or two check-ins per day/week/month).



Outline

- Worst case scenario for attackers (= reality?)…
  - No elements are observed in P_2 & P_3. → Cannot de-anonymize u_2 & u_3.



(ML: Maximum Likelihood Estimation)

Q. Is it possible to de-anonymize traces using such training data?

- Our Contributions
  - We show the answer is **"yes"**.
  - We propose a training method that outperforms a random guess even when no elements are observed in more than 70% of cases.

Contents

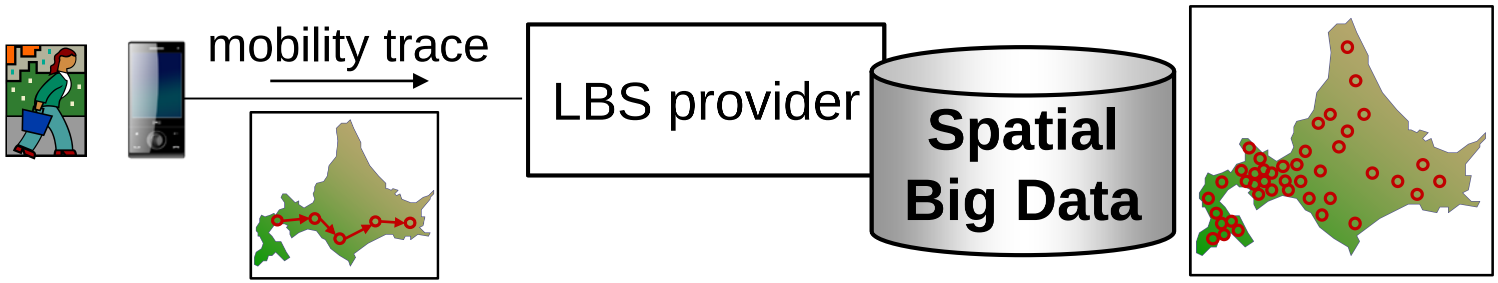
Introduction
(Location Privacy, Related Work)

Our Proposal
(EMTF: Expectation-Maximization Tensor Factorization)

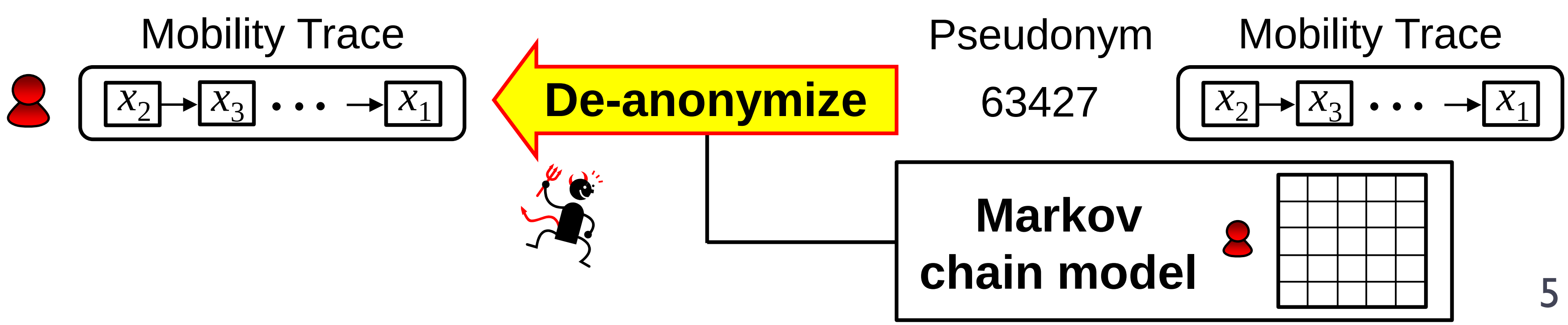
Experiments

Location Privacy

- Location-based Services (LBS)
  - Many people are using LBS (e.g. map, route finding, check-in).
  - “Spatial Big Data” can be provided to a third-party for analysis (e.g. popular places), or made public to provide traffic information.



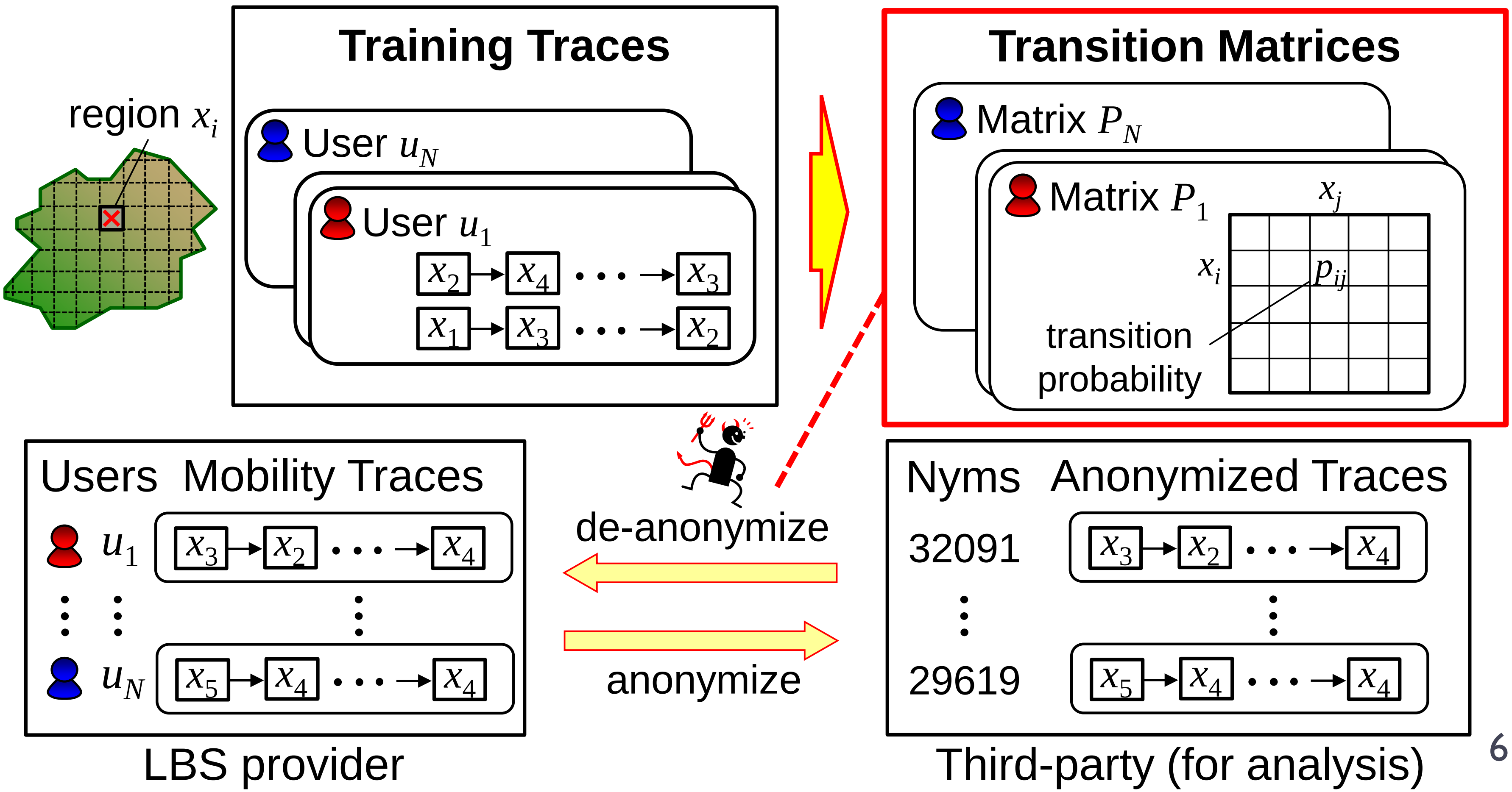
- Privacy Issues
  - Mobility trace can contain sensitive locations (e.g. homes, hospitals).
  - **Anonymized trace may be de-anonymized.**



Related Work

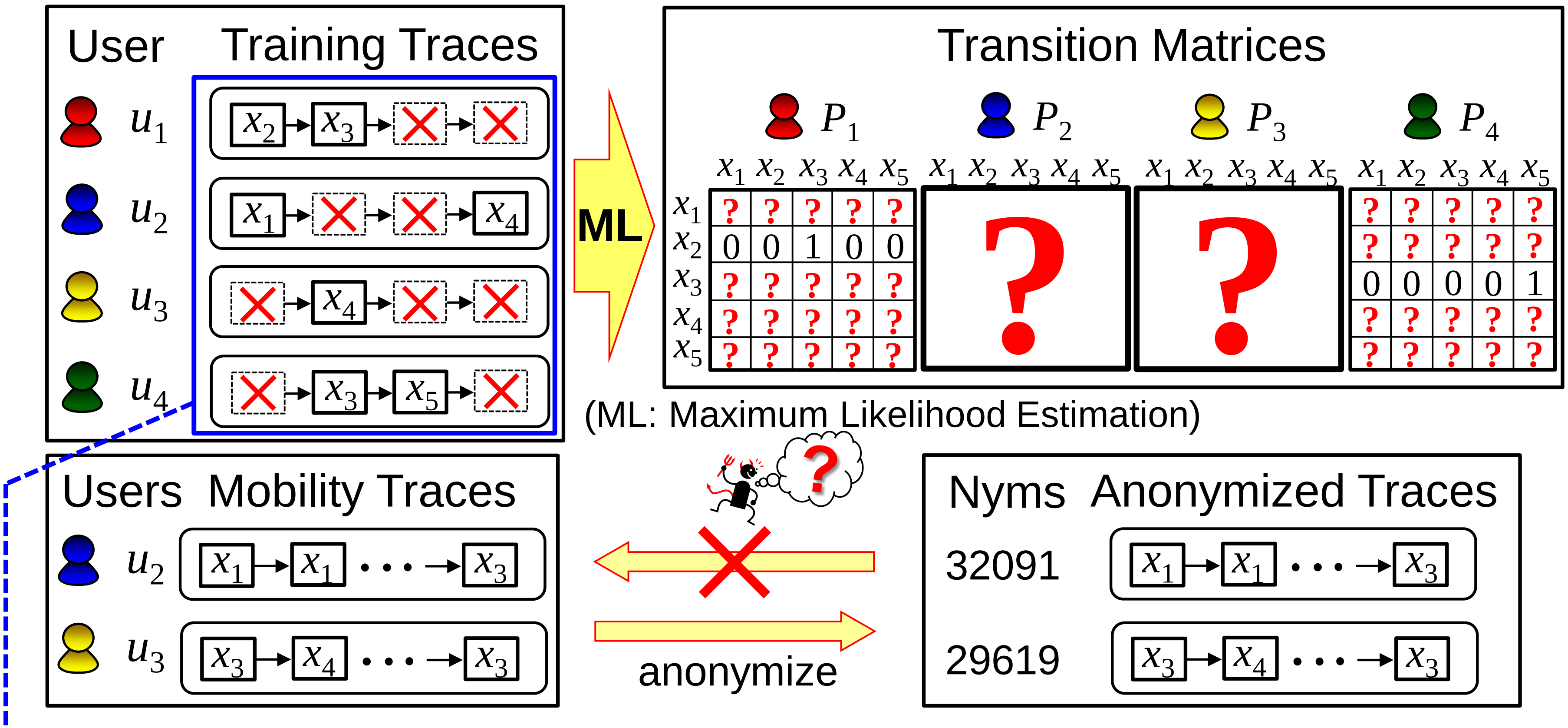
[Shokri+,S&P11] [Gambs+,JCSS14] [Mulder+,WPES08] etc.

- Markov Chain Model for De-anonymization
  - Attacker = anyone who has anonymized traces (except for LBS provider).
  - Attacker obtains training locations that are made public (e.g. via SNS).
  - Attacker de-anonymizes traces using the trained transition matrices.



Related Work

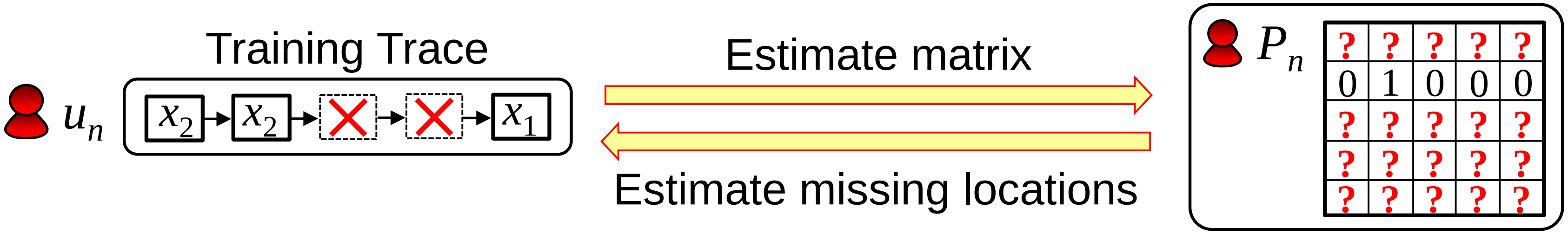
- Sporadic Training Data (training data are sparsely distributed over time)
  - Many users disclose a small number of locations “sporadically” (via SNS).
  - If we don’t estimate missing locations, we cannot train P_2 and P_3.
  - → we cannot de-anonymize traces of u_2 and u_3 using these matrices.



We need to “somehow” estimate missing locations.

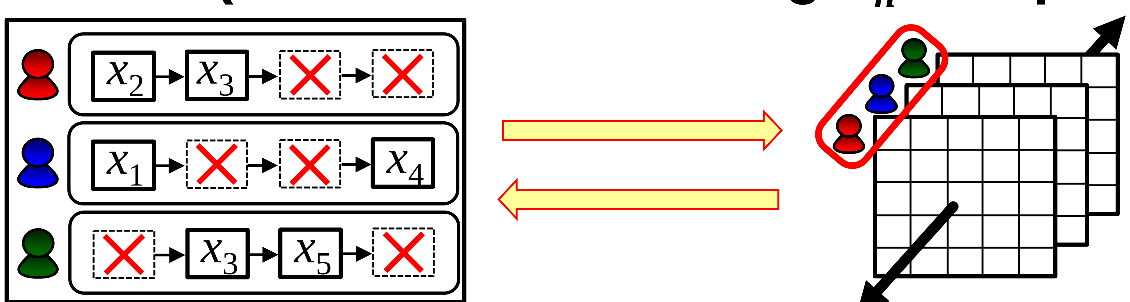
Related Work

- Gibbs Sampling Method [Shokri+, S&P11]
  - Alternates between estimating P_n and estimating missing locations of u_n independently of other users.



- Challenge
  - When there are few continuous locations in training traces...
  - (1) Cannot accurately estimate P_n.
  - (2) Cannot accurately estimate missing locations using P_n (→(1)).

We address this challenge by estimating P_n with the help of "other users" (instead of estimating P_n independently).



Contents

Introduction

(Location Privacy, Related Work)

Our Proposal

(EMTF: Expectation-Maximization Tensor Factorization)

Experiments

Overview of EMTF

We use the help of "similar users" (other users who have similar behavior):

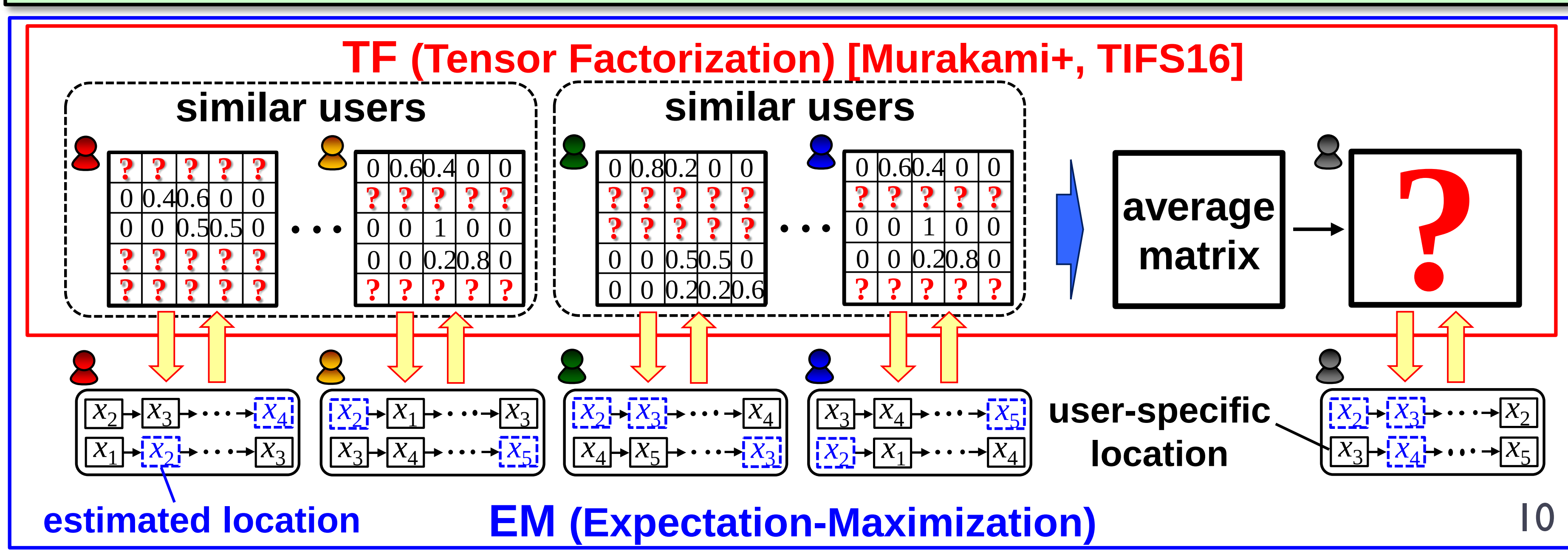
(1) Training Transition Matrices:

We estimate unobserved elements (**"?"**) with the help of **"similar users"**.
We substitute average matrix over all users for completely unobserved matrices.

(2) Estimating Missing Locations:

We estimate missing locations (we can do this with the help of **"similar users"**).

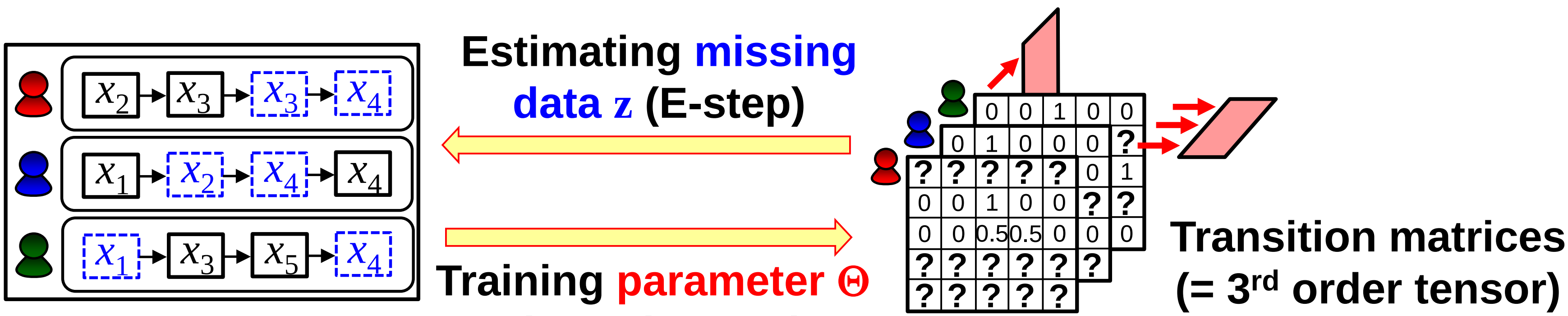
Go back to (1) → Each matrix captures **unique feature of each user's behavior** since each trace is accurate & user-specific.



Details of EMTF

- TF (Tensor Factorization)
  - Used for item recommendation. Factorizes tensor into low-rank matrices.
  - Estimates unobserved element (**“?”**) with the help of **“similar users”**.

- EM (Expectation-Maximization)
  - Trains **parameter Θ** from observed data $\mathbf{x}$ while estimating **missing data** $\mathbf{z}$.
  - Each EM cycle is guaranteed to increase the posterior probability $\Pr(\Theta|\mathrm{x})$.



$\mathbf{x} = (x_2, x_3, x_1, x_4, x_3, x_5)$

$\mathbf{z} = (x_3, x_4, x_2, x_4, x_1, x_4)$

Can find the most probable Θ and $\mathbf{z}$ with the help of “similar users”.

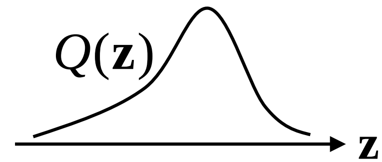
EMTF Algorithm

E-step: Estimate a distribution of missing location vector $\mathbf{z}$:

$$Q(\mathbf{z}) := \Pr(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\Theta})$$

Forward-Backward algorithm

M-step: Estimate parameter $\hat{\boldsymbol{\Theta}}$ in TF given by

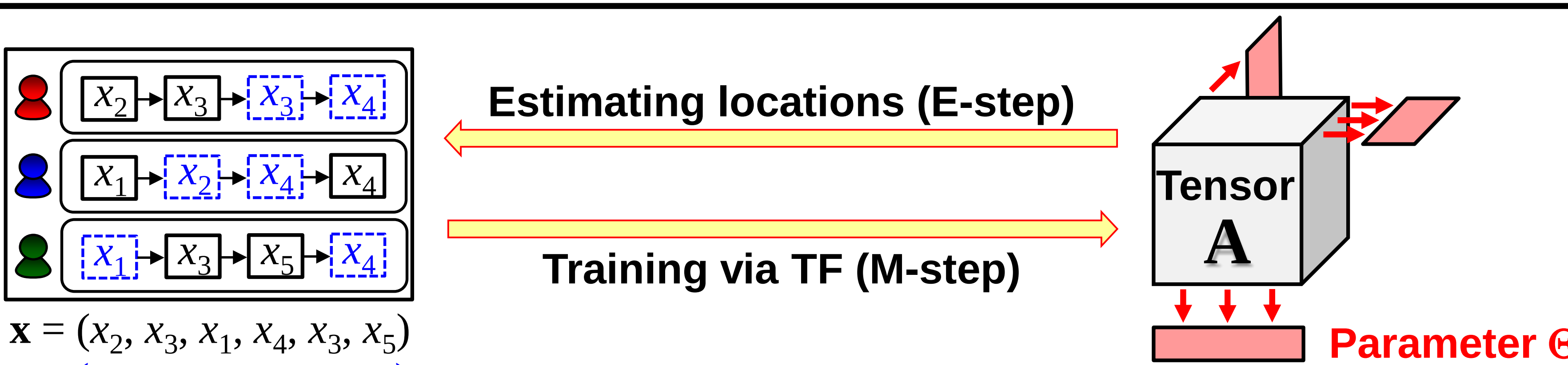


$$\hat{\boldsymbol{\Theta}} = \arg\max_{\Theta \geq 0} \sum_{\mathbf{z}} Q(\mathbf{z}) \underline{\log \Pr(\boldsymbol{\Theta} \mid \mathbf{x}, \mathbf{z})}$$

$$= \arg\min_{\Theta \geq 0} \sum_{\mathbf{z}} Q(\mathbf{z}) \underline{(\| \mathbf{A} - \hat{\mathbf{A}} \|_F^2 + \lambda \| \boldsymbol{\Theta} \|_F^2)}$$

Quadratic problem (w.r.t. one parameter)

Max of log-posterior = Min of regularized square error



$\mathbf{x} = (x_2, x_3, x_1, x_4, x_3, x_5)$

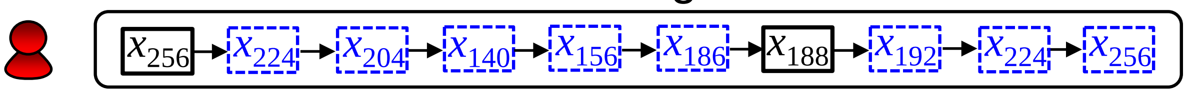
$\mathbf{z} = (x_3, x_4, x_2, x_4, x_1, x_4)$

Time complexity is exponential in the number of missing locations.☹

Approximation of EMTF

- Time Complexity of EMTF
  - Number of possible missing locations $\mathbf{z}$ is exponential in its length.
  - E.g. #(regions) = 256, #(missing locations) = 8 → possible $\mathbf{z}$ is $256^8 = 2^{64}$.

Training Trace

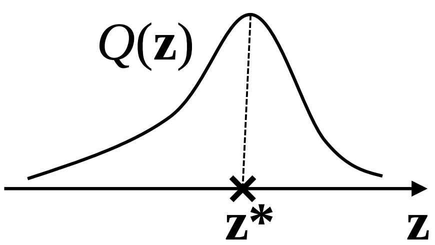


$\mathbf{z} = (x_{224}, x_{204}, x_{140}, x_{156}, x_{186}, x_{192}, x_{224}, x_{256})$

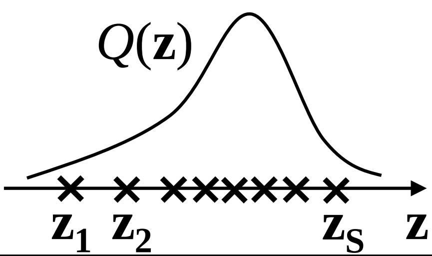
$Q(\mathbf{z})$ (distribution of $\mathbf{z}$)

- Two Approximation Methods:
  - **[Method I] Viterbi**: Approximates $Q(\mathbf{z})$ by the most probable value $\mathbf{z}^*$.
  - **[Method II] FFBS**: Approximates $Q(\mathbf{z})$ by random samples $\mathbf{z}_1, \ldots, \mathbf{z}_S$.

Viterbi



FFBS (Forward Filtering Backward Sampling)



Approximate Q(z) in a more accurate manner

Both methods reduce time complexity from exponential to linear.

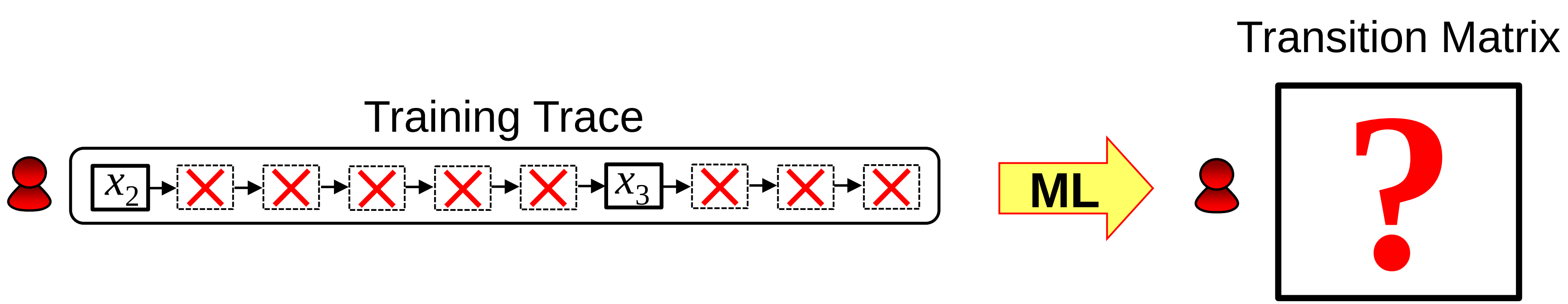
Contents

Experimental Set-up

(Here we explain only the most important part. Please see our paper for details)

- Gowalla Dataset
  - We used traces in New York & Philadelphia (16 x 16 regions).
  - **Training:** 250 users x 1 traces x 10 locations (time interval: more than 30min).
  - **Testing:** 250 users x 9 traces x 10 locations.
  - We randomly deleted each training location with probability 80%.
  - → No elements in a matrix were observed in **more than 70% of cases**.

Extremely Sporadic Training Data (Worst Case Scenario for Attackers)

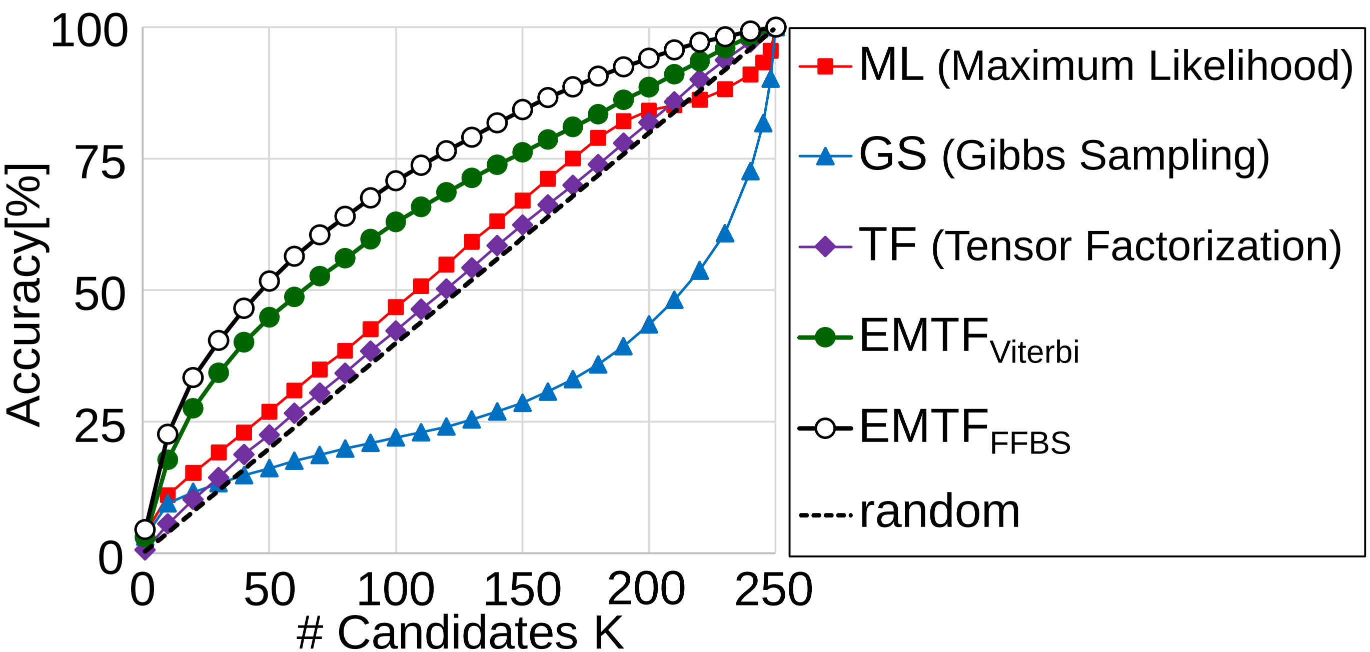


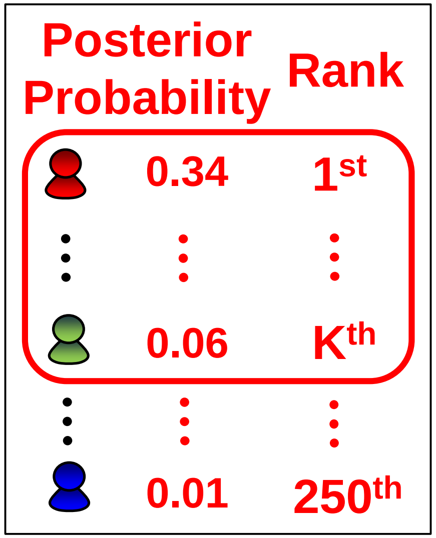
More than 70% of cases

(ML: Maximum Likelihood Estimation)

Experimental Results

- De-anonymization Accuracy
  - We performed the Bayesian de-anonymization attack, which selects, for each testing trace, K (<250) candidates whose probabilities are the highest.
  - ML & TF ≈ random guess
    - since they did not estimate missing locations.
  - GS < random guess
    - since it did not accurately estimate missing locations.

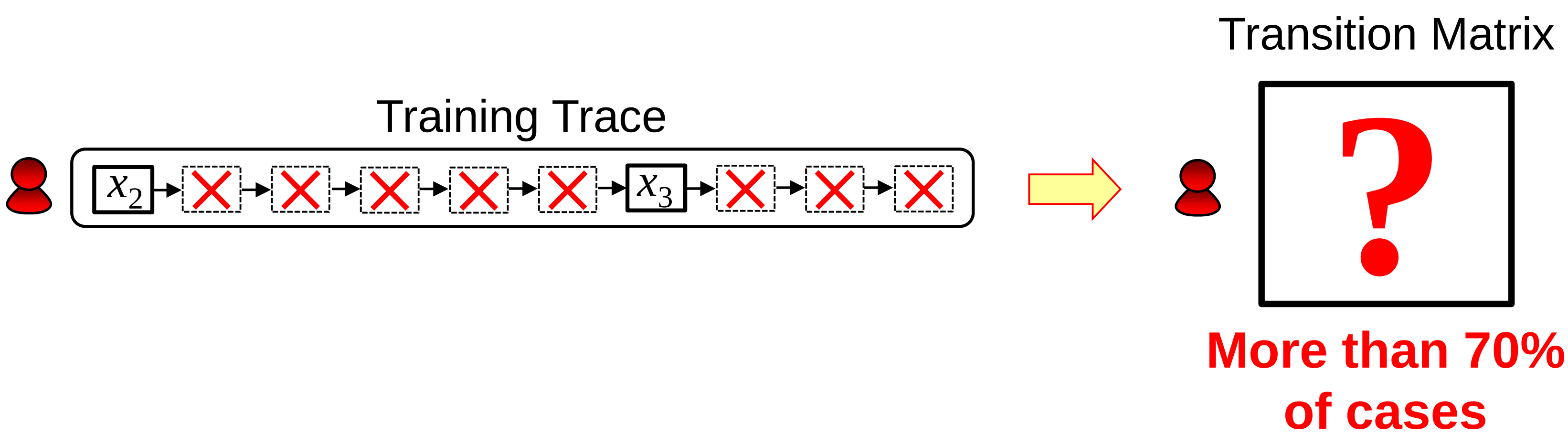




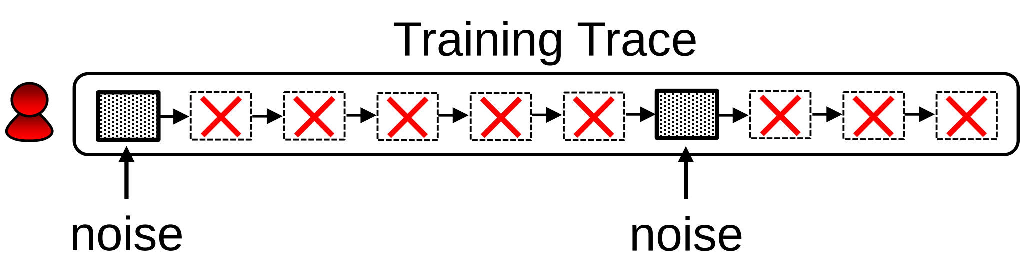
EMTF outperformed random guess in sporadic training data scenario.

Conclusion

- Summary of Results
  - Our training method (EMTF) significantly outperformed a random guess, even when no elements were observed in more than 70% of cases.



- Future Work
  - Evaluation of state-of-the-art obfuscation (e.g. geo-indistinguishability [Andres+, CCS13]) applied to sporadic training traces.

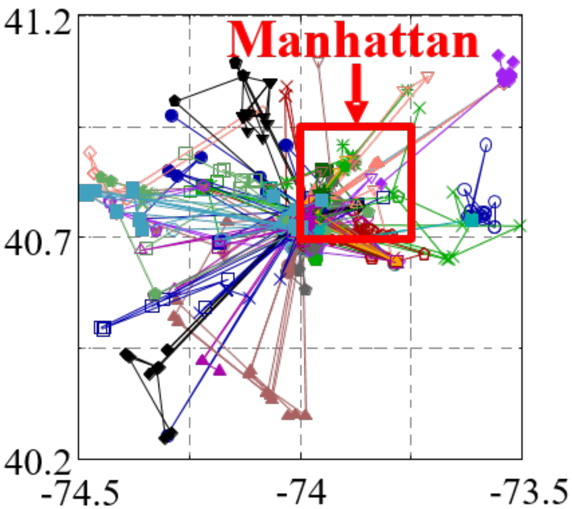


Thank you for listening.

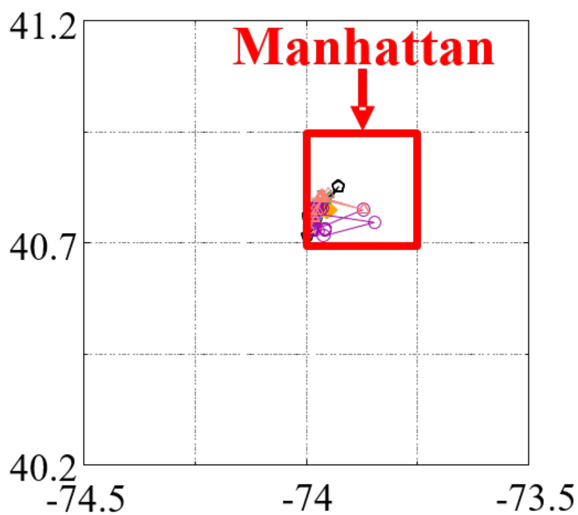
Appendix: Similar Users in Gowalla Dataset

- TF (Tensor Factorization)
  - Can automatically find a set of users who have “similar behavior”.
  - Trains matrices so that each matrix is influenced by similar users.

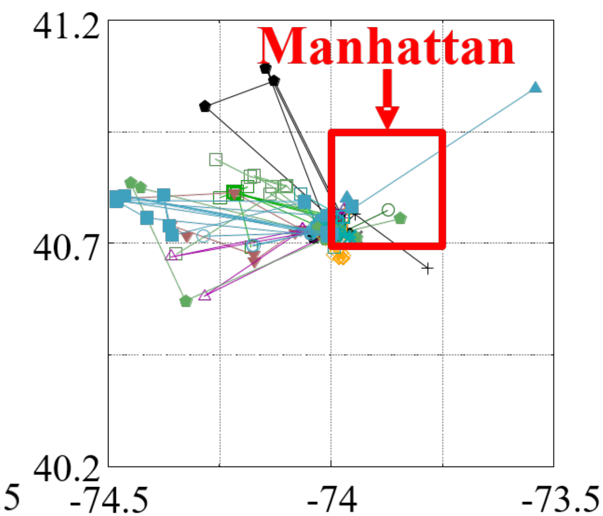
- Visualization of “similar users” [Murakami+, TIFS16]
  - We visualized “similar users” in Gowalla based on the trained parameters.
  - E.g. always stay in Manhattan, go to the western part of Manhattan.



All Users



Users who had a large value in 1st parameter



Users who had a large value in 2nd parameter.