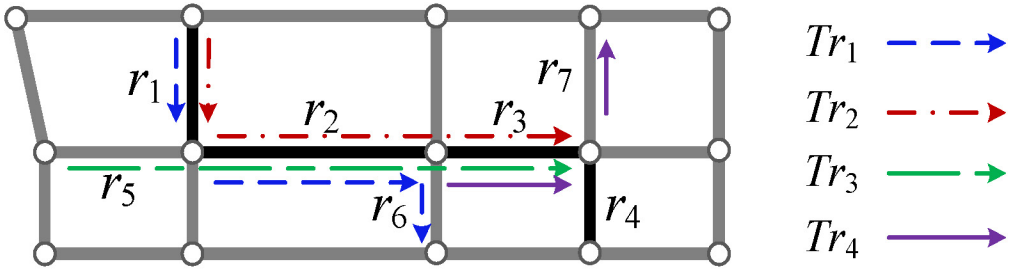
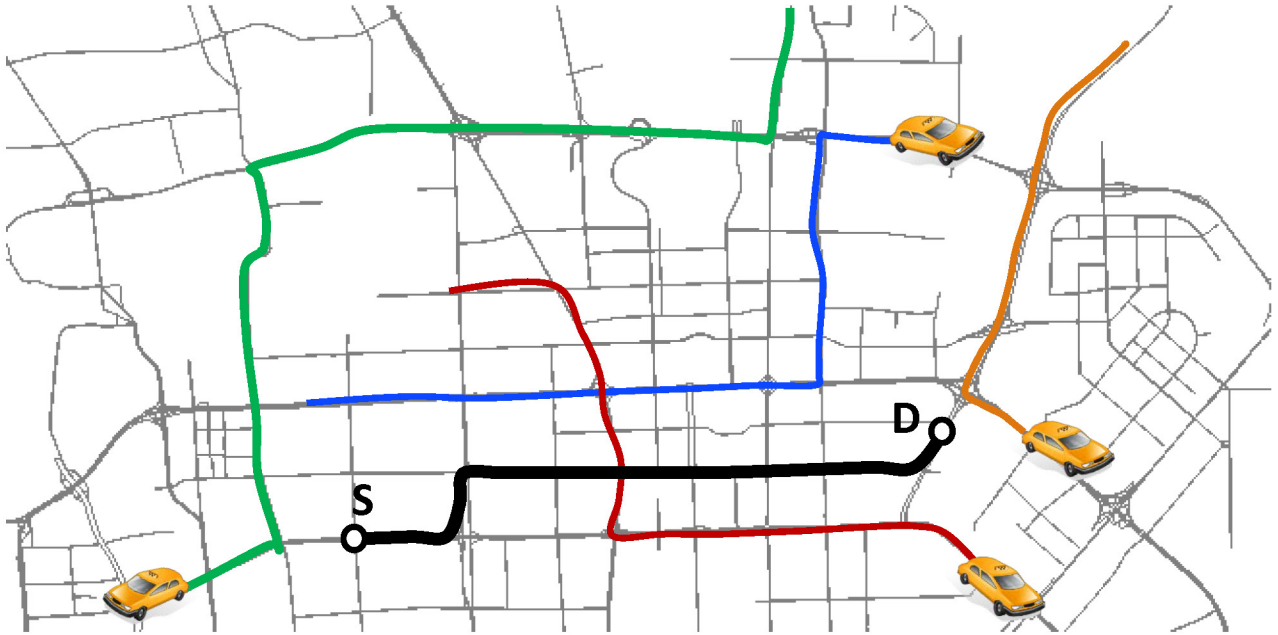
Travel Time Estimation of a Path using Sparse Trajectories

Dr. **Yu Zheng** yuzheng@microsoft.com
Lead Researcher, Microsoft Research
Chair Professor at Shanghai Jiao Tong University



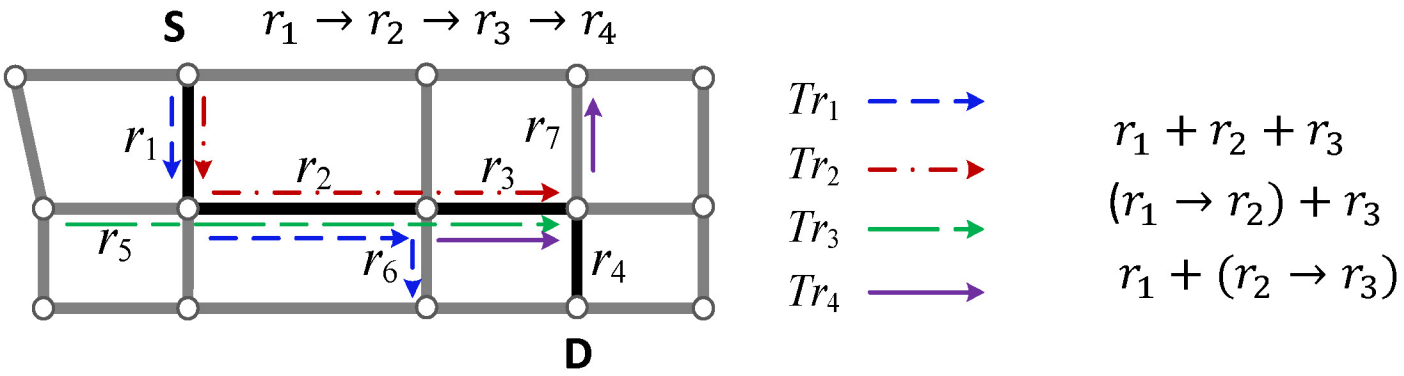
Goal

- Estimate the travel time of any given path
  - on road network instantly
  - using historical and current trajectories generated by a sample of vehicles



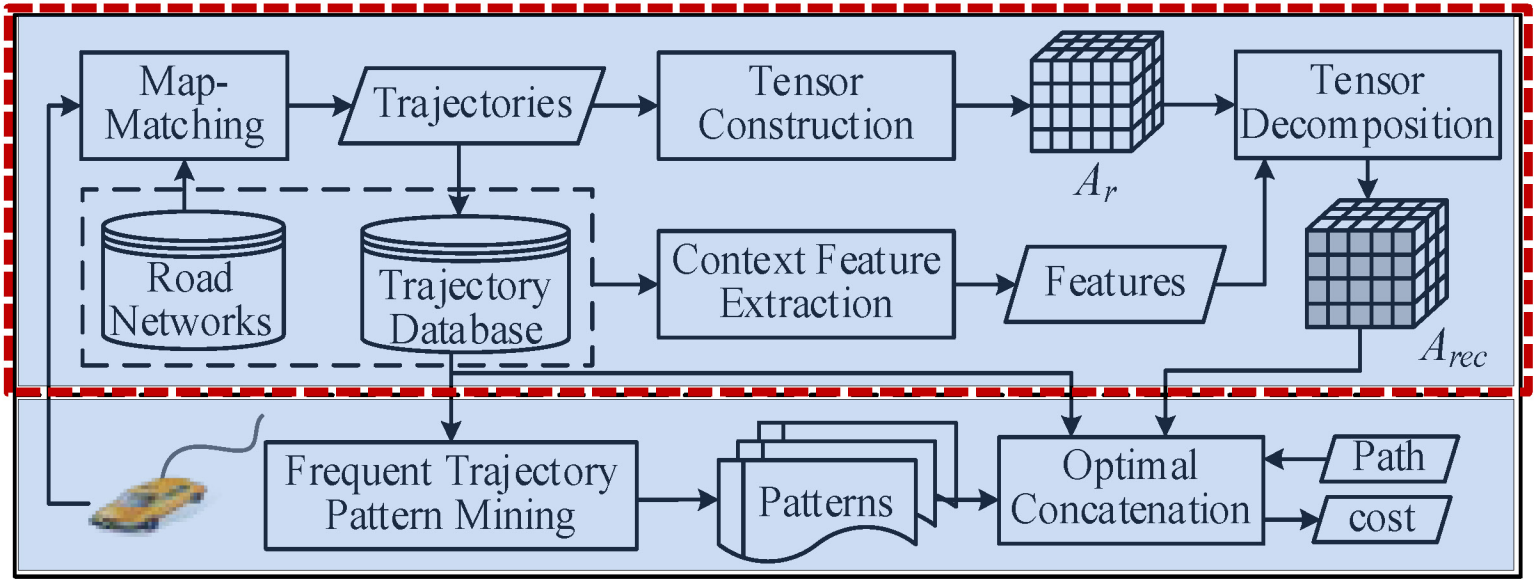
Challenges

- Data sparsity
- Trajectory concatenation
  - Multiple ways to combine sub-trajectories
  - Length of a sub-trajectory and its support
- Scalability and efficiency
  - A citywide estimation
  - E.g. Beijing has over 100,000 road segments



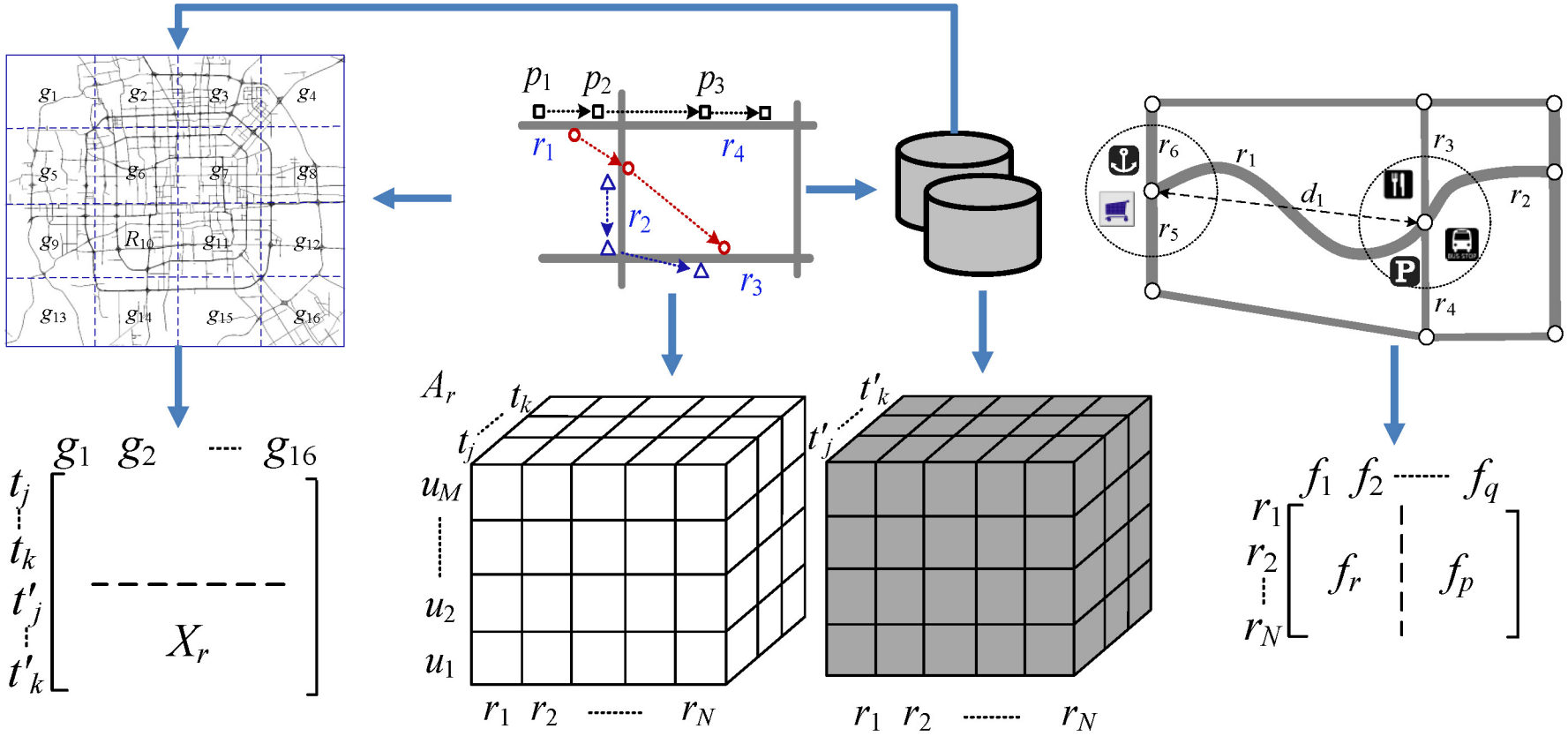
Methodology

- Framework
  - Context-Aware Tensor Decomposition (CATD)
  - Optimal Concatenation (OC)



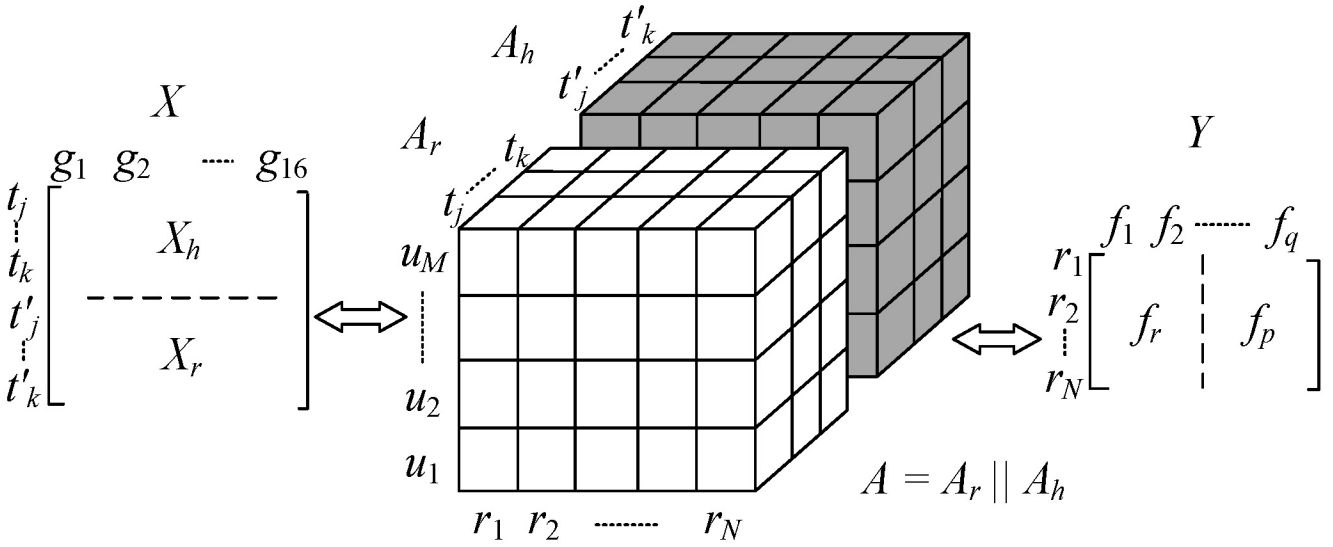
Methodology

- Supplementing missing values



Methodology

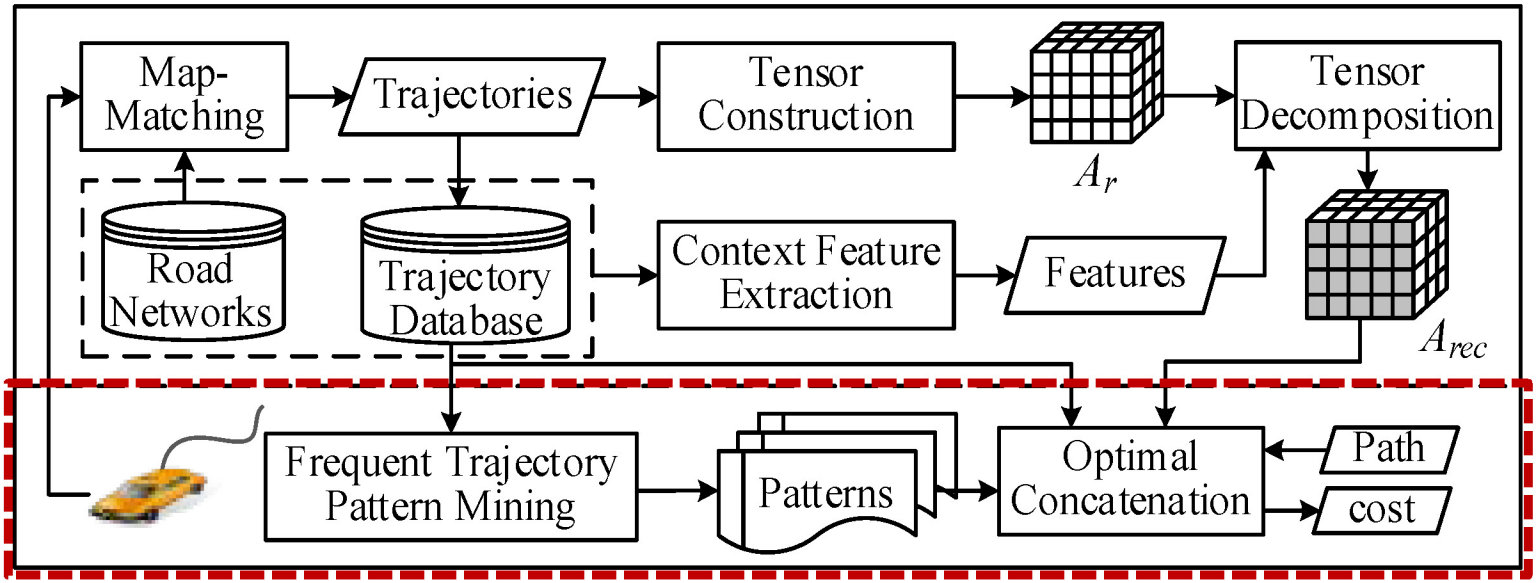
- Supplementing missing values



$$\mathcal{L}(S,R,U,T,F,G) = \frac{1}{2}\|\mathcal{A} - S \times_R R \times_U U \times_T T\|^2 + \frac{\lambda_1}{2}\|X - TG\|^2 + \frac{\lambda_2}{2}\|Y - RF\|^2 + \frac{\lambda_3}{2}(\|S\|^2 + \|R\|^2 + \|U\|^2 + \|T\|^2 + \|F\|^2 + \|G\|^2)$$

Methodology

- Framework



Methodology

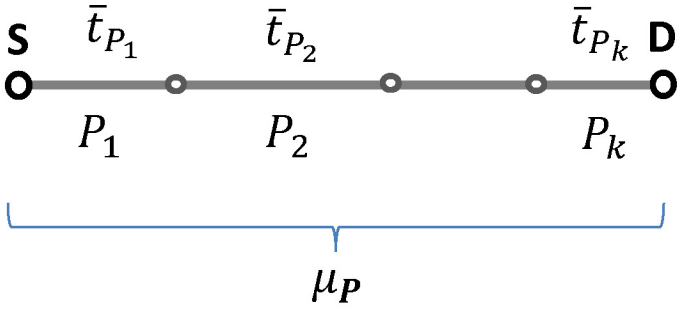
- Optimal Concatenation

Suppose $\boldsymbol{P}$ is decomposed as $P_1||P_2||\cdots||P_k$

true travel time: $\mu_{\boldsymbol{P}}$. estimated travel time: $\bar{t}_{P_1}+\bar{t}_{P_2}+\cdots+\bar{t}_{P_k}$

$$LSE_{\boldsymbol{P},P_1,P_2,\cdots,P_k} \triangleq E\left(\mu_{\boldsymbol{P}}-\bar{t}_{P_1}-\bar{t}_{P_2}-\cdots-\bar{t}_{P_k}\right)^2$$

$$\text{argmin}_{P_1,P_2,\cdots,P_k}\ LSE_{\boldsymbol{P},P_1,P_2,\cdots,P_k},$$
$$\text{subject to}\ \ P_1||P_2||\cdots||P_k=\boldsymbol{P}$$



Methodology

- Optimal Concatenation

$$LSE_{\boldsymbol{P},P_1,P_2,\cdots,P_k} = E\left(\mu_{\boldsymbol{P}} - \bar{t}_{P_1} - \bar{t}_{P_2} - \cdots - \bar{t}_{P_k}\right)^2$$

$$= E\left(\mu_{P_1} + \mu_{P_2} + \cdots \mu_{P_k} - \bar{t}_{P_1} - \bar{t}_{P_2} - \cdots - \bar{t}_{P_k}\right)^2$$

$$= E\left(\sum_{i=1}^{k}\left(\mu_{P_i} - \bar{t}_{P_i}\right)^2 + \sum_{i=1}^{k}\sum_{j=1}^{k}(\mu_{P_i} - \bar{t}_{P_i})(\mu_{P_j} - \bar{t}_{P_j})\right)$$

$$= \sum_{i=1}^{k} E(\mu_{P_i} - \bar{t}_{P_i})^2 + \sum_{i=1}^{k}\sum_{j=1}^{k} E\left((\mu_{P_i} - \bar{t}_{P_i})(\mu_{P_j} - \bar{t}_{P_j})\right)$$

assuming $\bar{t}_{P_i}$ and $\bar{t}_{P_j}$ are independent

$$E\left((\mu_{P_i} - \bar{t}_{P_i})(\mu_{P_j} - \bar{t}_{P_j})\right) = E\left(\mu_{P_i} - \bar{t}_{P_i}\right)E(\mu_{P_j} - \bar{t}_{P_j}) = 0$$

$$LSE_{\boldsymbol{P},P_1,P_2,\cdots,P_k} = \sum_{i=1}^{k} E\left(\mu_{P_i} - \bar{t}_{P_i}\right)^2$$

$$E(\mu_{P_i} - \bar{t}_{P_i})^2 = E(\mu_{P_i} - \frac{1}{n_{P_i}}\sum_{j=1}^{n_{P_i}} t_{P_i,j})^2 = \frac{1}{n_{P_i}^2} E\sum_{j=1}^{n_{P_i}}(\mu_{P_i} - t_{P_i,j})^2$$

$$= \frac{1}{n_{P_i}^2}\sum_{j=1}^{n_{P_i}} E(\mu_{P_i} - t_{P_i,j})^2 = \frac{1}{n_{P_i}} Var(t_{P_i,j})$$

Methodology

- Support n_{P_i} vs. Variance $Var(t_{P_i,j})$
  - The bigger the support is, the smaller the error is
  - The bigger the variance the bigger the error

$$\text{argmin}_{P_1,P_2,\cdots,P_k} \sum_{i=1}^{k} \frac{1}{n_{P_i}} Var(t_{P_i,j})$$

$$\text{subject to } P_1||P_2||\cdots||P_k = \boldsymbol{P}$$

- Solved by dynamic programming
  - Denote $g(P_i) = \frac{1}{n_{P_i}} Var(t_{P_i,j})$

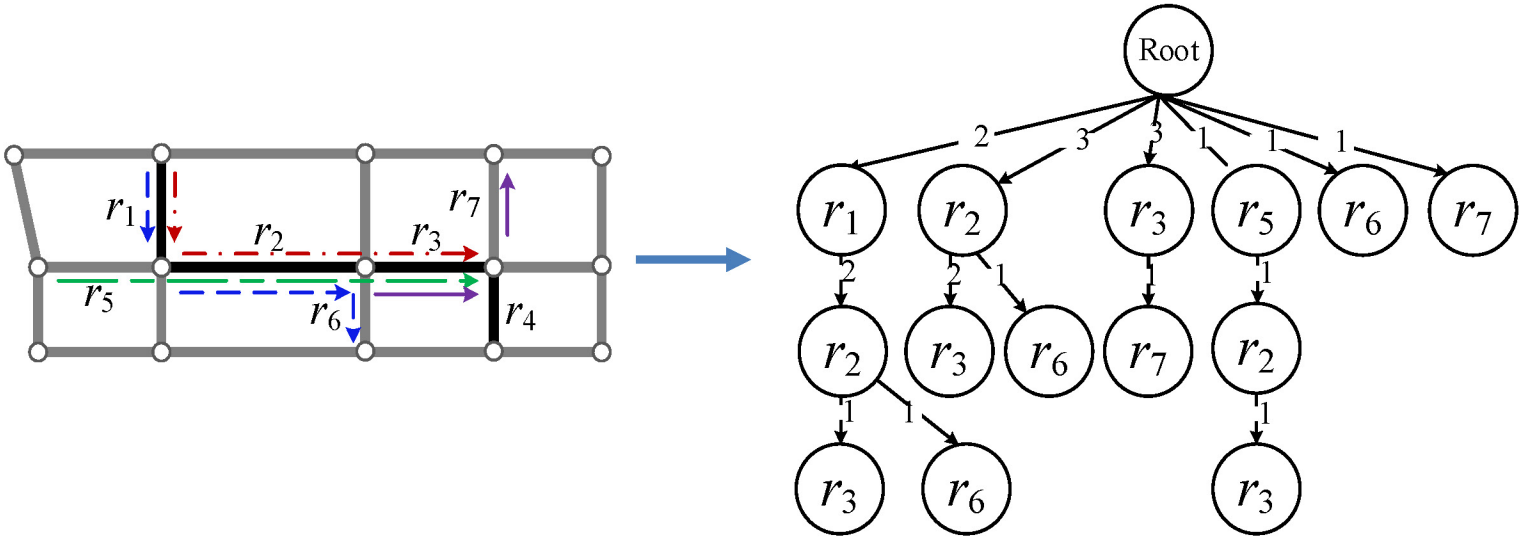
$$\text{argmin}_{P_1,P_2,\cdots,P_l} \sum_{j=1}^{l} g(P_j)$$

$$\text{subject to } P_1||P_2||\cdots||P_l = P'.$$

$$opt_n = \min_{1 \le i < n} (opt_i + g(P_{r_{i+1}||r_{i+2}\cdots||r_n}))$$

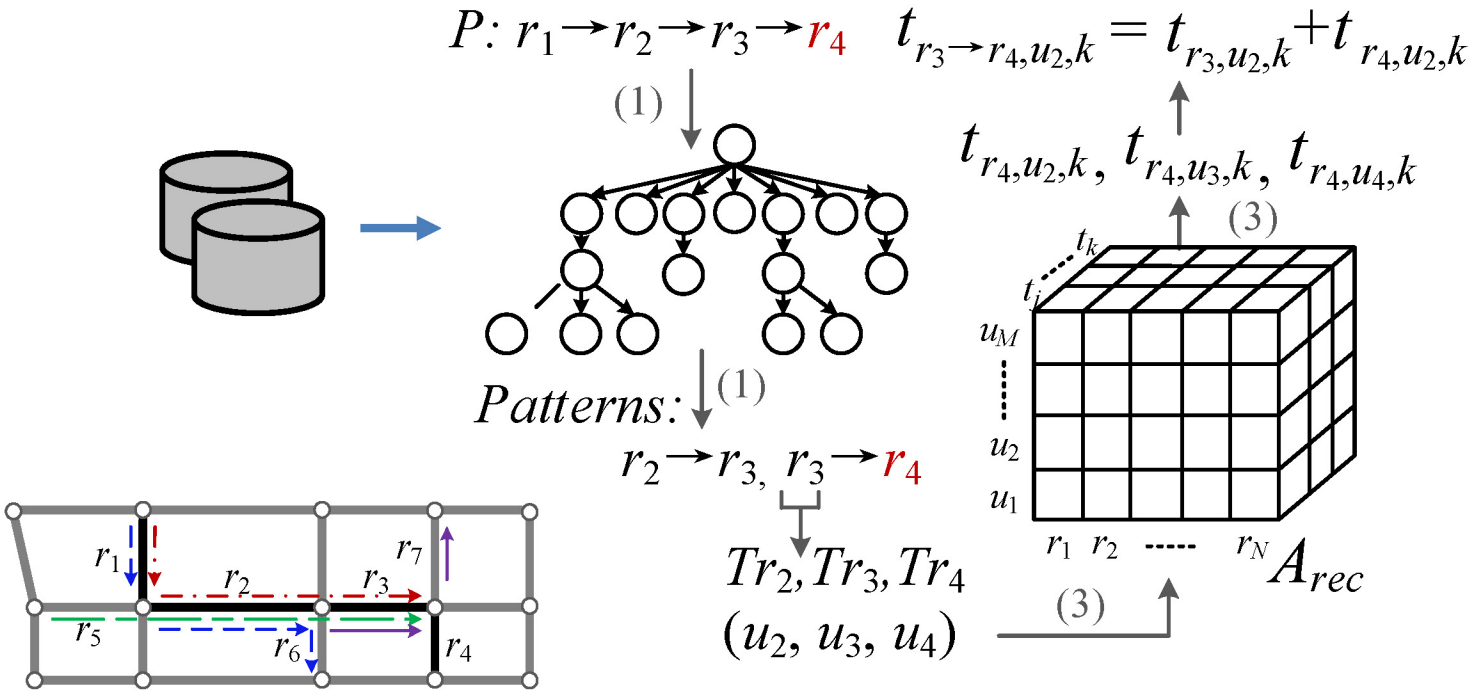
Methodology

- Make optimal concatenation more efficient
- Frequent trajectory pattern mining
  - Not necessary to check every combination
  - Using suffix-tree-based method



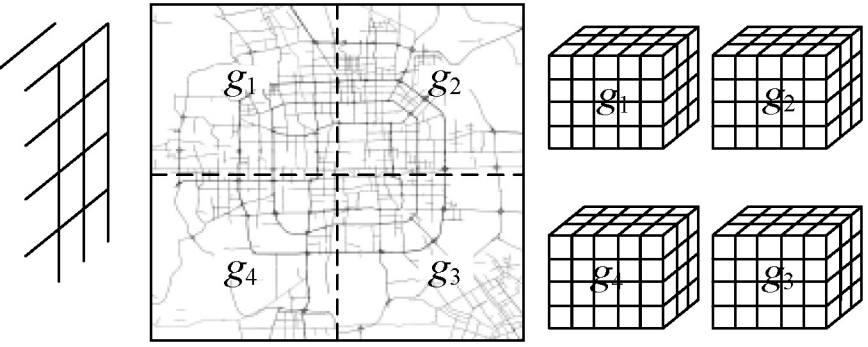
Methodology

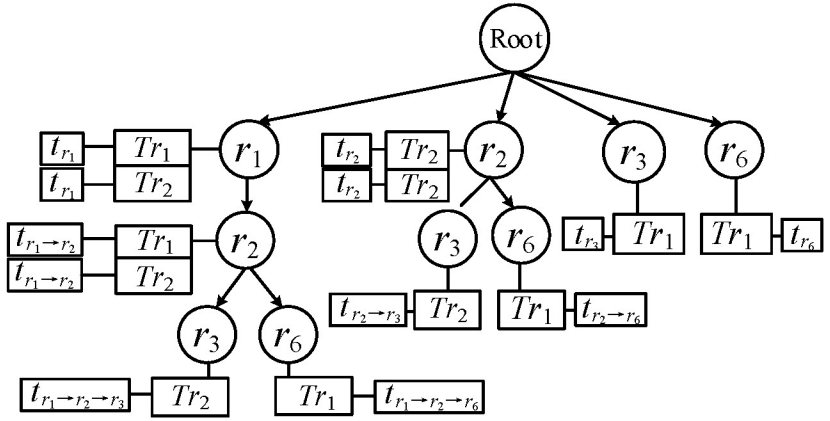
- Combining the suffix tree with tensors
  - Searching for frequent trajectory pattern from the suffix tree
  - Find the travel time of a particular user from the tensor



Methodology

- Deal with efficiency and scalability
  - data-driven space partition
  - an element-wise optimization algorithm for TD
  - Use trajectory patterns as concatenation candidates
  - Indexing recent trajectories for a fast online retrieval





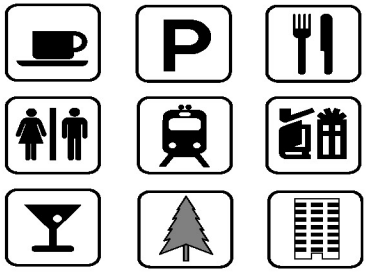
Experiments

[Download data here]

- Datasets
  - *Taxi Trajectories:*
    - Generated by 32,670 taxicabs in Beijing
    - From Sept. 1 to Oct. 31, 2013.
    - 673+ million GPS points
    - Total length: over 26 million km.
    - Sampling rate: 96 seconds per point.

  - *Road networks*:
    - 148,110 nodes and 196,307 edges.
    - Covers a 40×50km spatial range
    - Total length of road segments: 21,985km.

  - *POIs*:
    - Th273,165 POIs of Beijing
    - 195 tier two categories.
    - Chose the top 10 categories that occur around road segments



Experiments

- Performance of CATD
  - Remove 30% non-zero entries
  - $\mathcal{A}_r$/(5×5): 4,736×12,674×4; 0.09%

	MAE (min)	RMSE
TD	0.747	1.646
$TD + H$	0.732	1.629
$CATD\ (TD + H + C)$	0.714	1.613

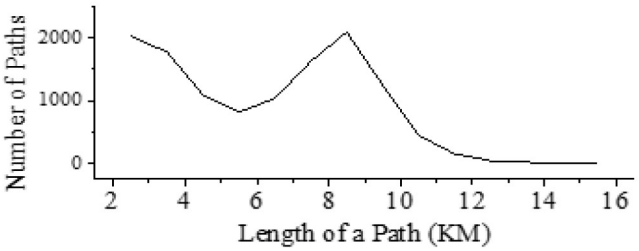
Metrics

- $MAE = \frac{\sum_i |y_i - \widehat{y_i}|}{n}$
- $MRE = \frac{\sum_i |y_i - \widehat{y_i}|}{\sum_i \widehat{y_i}}$
- $RMSE = \sqrt{\frac{\sum_i (y_i - \widehat{y_i})^2}{n}}$

Experiments

- Query paths
  - From taxi trajectories
    - 12,384 queries, 50 paths per hour/day
    - Traveled by at least two drivers
    - Total length 76,412.6km
    - Effective time span: 2,734 hours
  - In the field study
    - 114 queries
    - from Sept. 1 to Oct. 30, 2013
    - Total length 999km
    - Effective time span: 62 hours

A) Geographical distribution



B) Distribution of the length

C) Distribution of time length

Experiments

- Baselines
  - *Speed-Constraint-based* (SC) method
  - *Trajectory-based Simple Concatenation* (TSC) method.
  - *Optimal Concatenation with Historical Travel Time* (OC+H) method.
  - *Optimal Concatenation with Nonnegative Matrix Factorization* (OC+MF)

Query paths from taxi data

	MAE (min)	MRE	MAE/L (min/km)
SC	8.808	0.665	1.428
TSC	5.244	0.396	0.850
$OC + H$	3.245	0.245	0.526
$OC + MF$	3.061	0.231	0.496
$\boldsymbol{PTTE}$	**2.545**	**0.192**	**0.412**

In-the-field study

	MAE (min)	MRE	MAE/L (min/km)
SC	18.193	0.561	2.075
TSC	11.300	0.349	1.289
$OC + H$	4.990	0.154	0.569
$OC + MF$	4.052	0.125	0.462
$\boldsymbol{PTTE}$	**3.771**	**0.116**	**0.430**

Experiments

- Efficiency
  - 30 minutes per time slot
  - Infer the travel time of a path in **2.3ms**

	Components		Time	Memory (MB)
Deal with missing values ($CATD$)	Building matrix X, Y		34ms	9
	Tensor construction	$\mathcal{A}_r$	44ms	4.4
		$\mathcal{A}_h$	233ms	14.6
	Decomposition	5×5	6.31min	116
	Total		6.4min	144
Optimal Concatenation (OC)	Best OC		2.3ms	995
	w/o trajectory patterns		8.6ms	877
	w/o index		12.2s	714

Conclusion

- A very fundamental but challenging task
  - Data sparsity
  - Trade off between length of a path and support of trajectories
  - Efficiency and scalability
- Our method
  - Context-Aware Tensor Decomposition
  - Optimal Concatenation
- Results
  - Effectiveness
    - Relative error ratio: 19% and 11.6%
  - Efficiency
    - Infer the travel time of a path in **2.3ms**

Download data and codes

Search for “Urban Computing”

Thanks!

Yu Zheng

yuzheng@microsoft.com

Homepage