Trajectory analysis

Ivan Kukanov

UNIVERSITY OF EASTERN FINLAND

Joensuu, 2014

Semantic Trajectory Mining for Location Prediction

Josh Jia-Ching Ying

Tz-Chiao Weng

Vincent S. Tseng

Taiwan

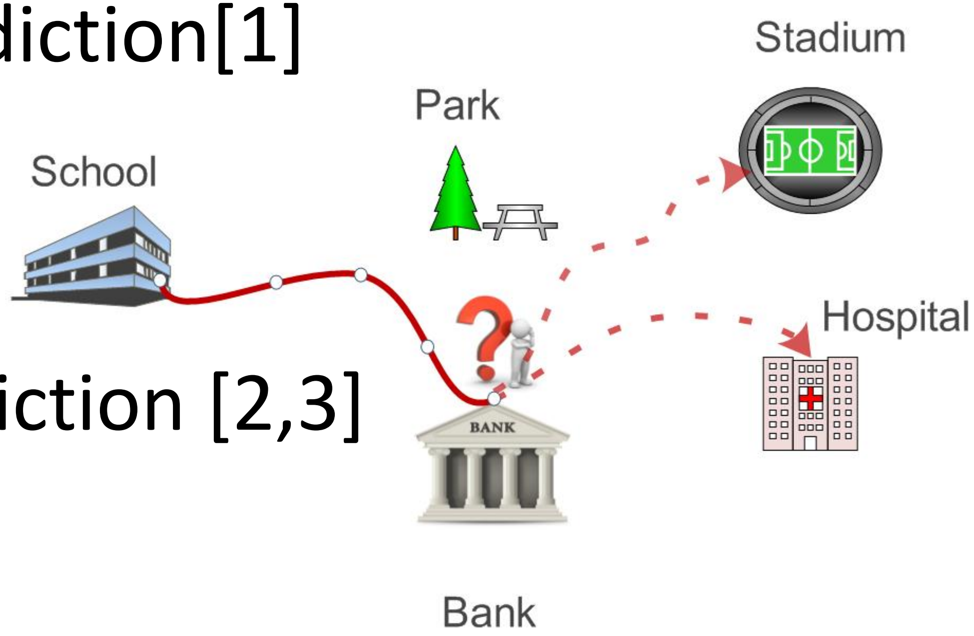
Wang-Chien Lee

Wang-Chien Lee

USA

The next location prediction approaches

- Personal-based prediction[1]
- General-based prediction [2,3]



Note: further consider General-based prediction

Semantic Trajectory Mining for Location Prediction

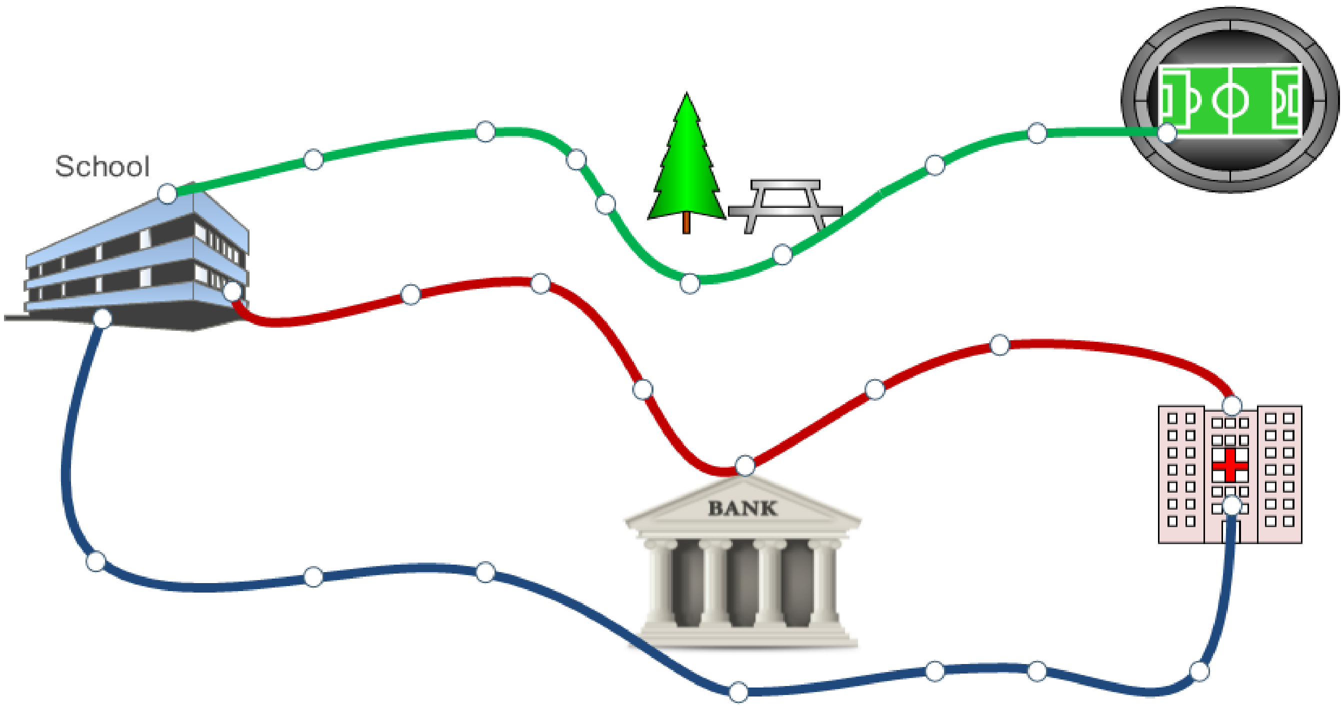
- predicting the next location of a user's movement

Input:

- GPS user's trajectories

Geographical and Semantic similarity

- Example: T_1 and T_2 – geograficaly,
  T_2 and T_3 – semantically



SemanPredict framework

- *Offline* and *Online* Modules
- Novel prediction strategy: *Semantic* and *Geographic* trajectory patterns
- MIT reality dataset [4]
  - 106 mobile users
  - Over 500,000 hours

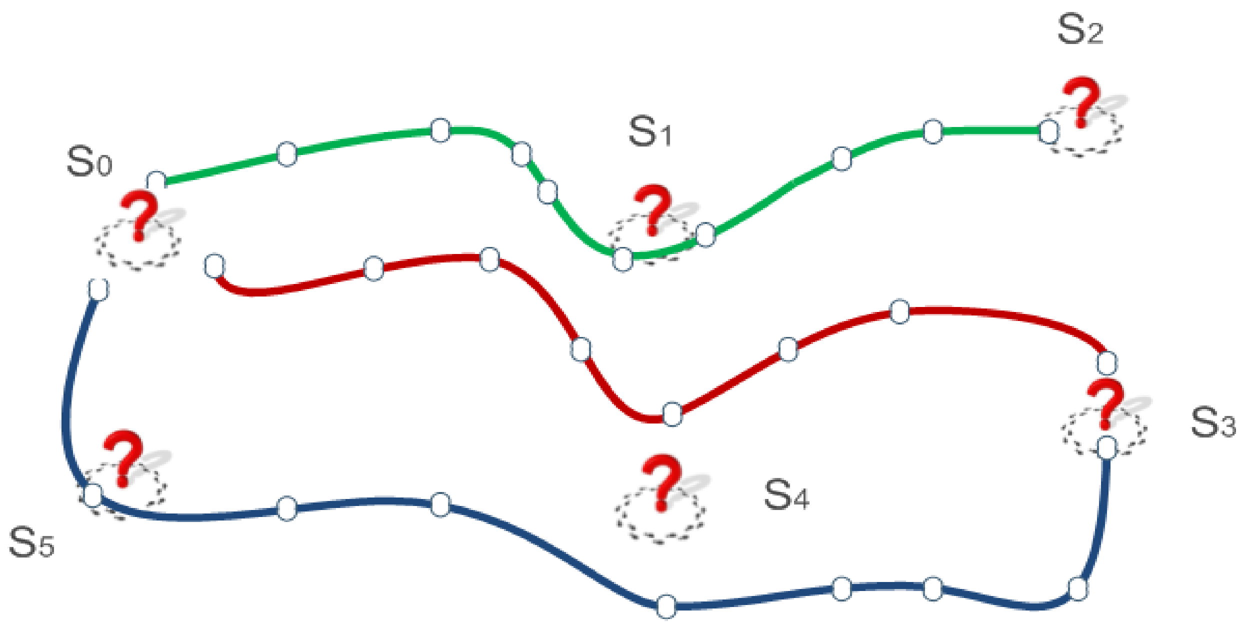
SemanPredict: offline module

1. Data preprocessing
2. Semantic mining
3. Geographic mining

SemanPredict: offline module

1. Data preprocessing:

GPS trajectories to *stay locations sequence* [1, 5]



Output: **T_1**=<S_0, S_1, S_2> ; **T_2**=<S_0, S_4, S_3>; **T_3**=<S_0, S_5, S_4, S_3>; ...

SemanPredict: offline module

2. Semantic mining:

2.1. Assign *semantic labels* to stay locations using gazetteer [6];

- Example:

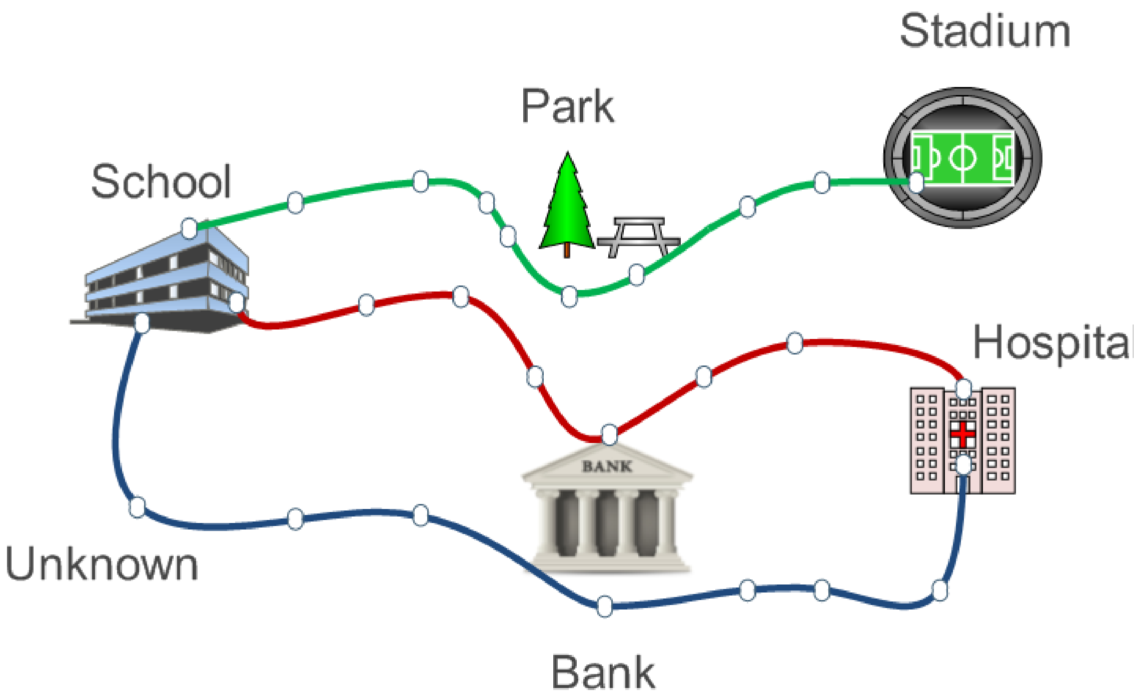
Input: $\mathbf{T_1}$=<S_0, S_1, S_2> ; $\mathbf{T_2}$=<S_0, S_4, S_3>; $\mathbf{T_3}$=<S_0, S_5, S_4, S_3>; …

Output: $\mathbf{T_1}$=<School, Park, Stadium> ; $\mathbf{T_2}$=<School, Bank, Hospital>; $\mathbf{T_3}$=<School, Unknown, Bank, Hospital>; …

SemanPredict: offline module

2. Semantic mining:

- Example: result *semantic labels*



SemanPredict: offline module

2. Semantic mining

2.2. Similar User Clustering

- Complete linkage clustering

Similarity measure: *Maximal Semantic Trajectory Pattern Similarity* [6]

SemanPredict: offline module

- Example: Longest Common Sequence(LCS)

P = <School, Park, Stadium>, Q = <School, Bank, Park, Shop>

LCS(P, Q) =<School, Park>

$$ratio(P) = \frac{1+1}{3} = \frac{2}{3} \qquad ratio(Q) = \frac{1+1}{4} = \frac{1}{2}$$

$$dist(P,Q) = \frac{1}{2} \cdot \left(\frac{2}{3} + \frac{1}{2} \right) = \frac{7}{12}$$

SemanPredict: offline module

3. Geographic mining

Input:

- stay location sequence
- clusters from **semantic mining**

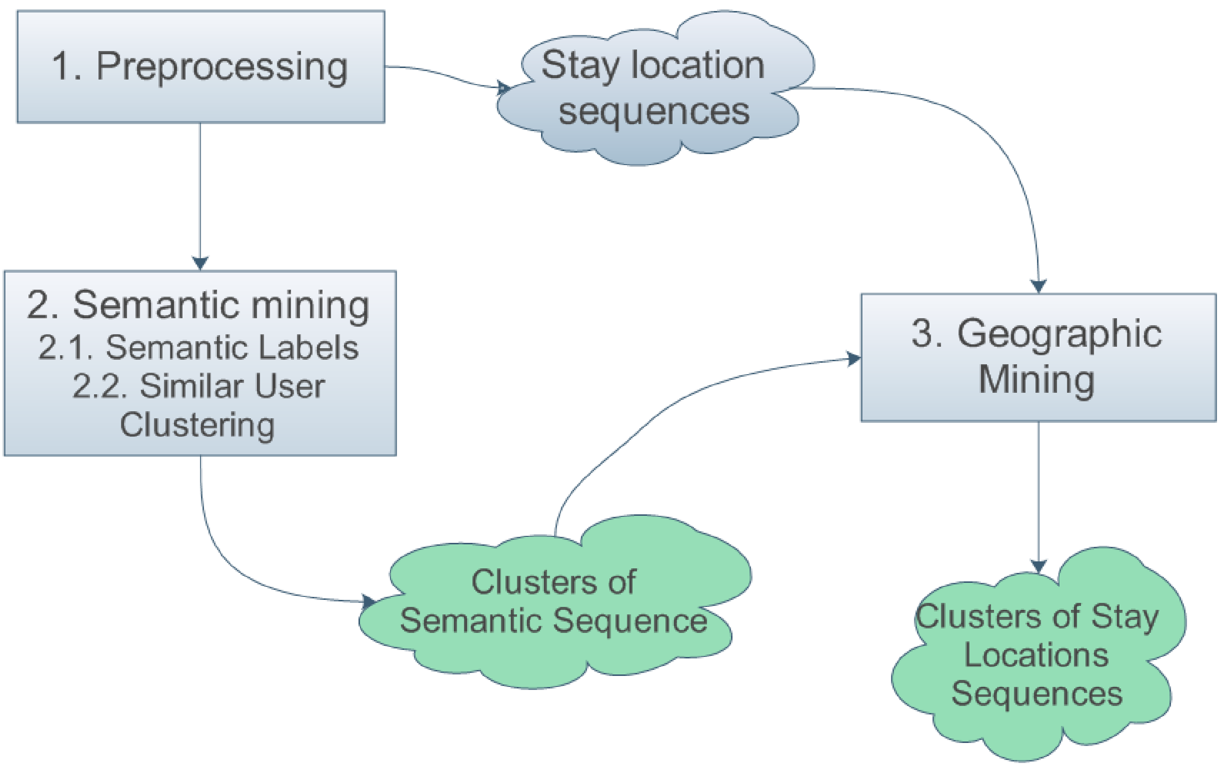
3.1. Grouping Users' Stay Location Sequences (*Prefix-Span* [9])

Output: clusters of stay location sequence based on **semantic clusters**

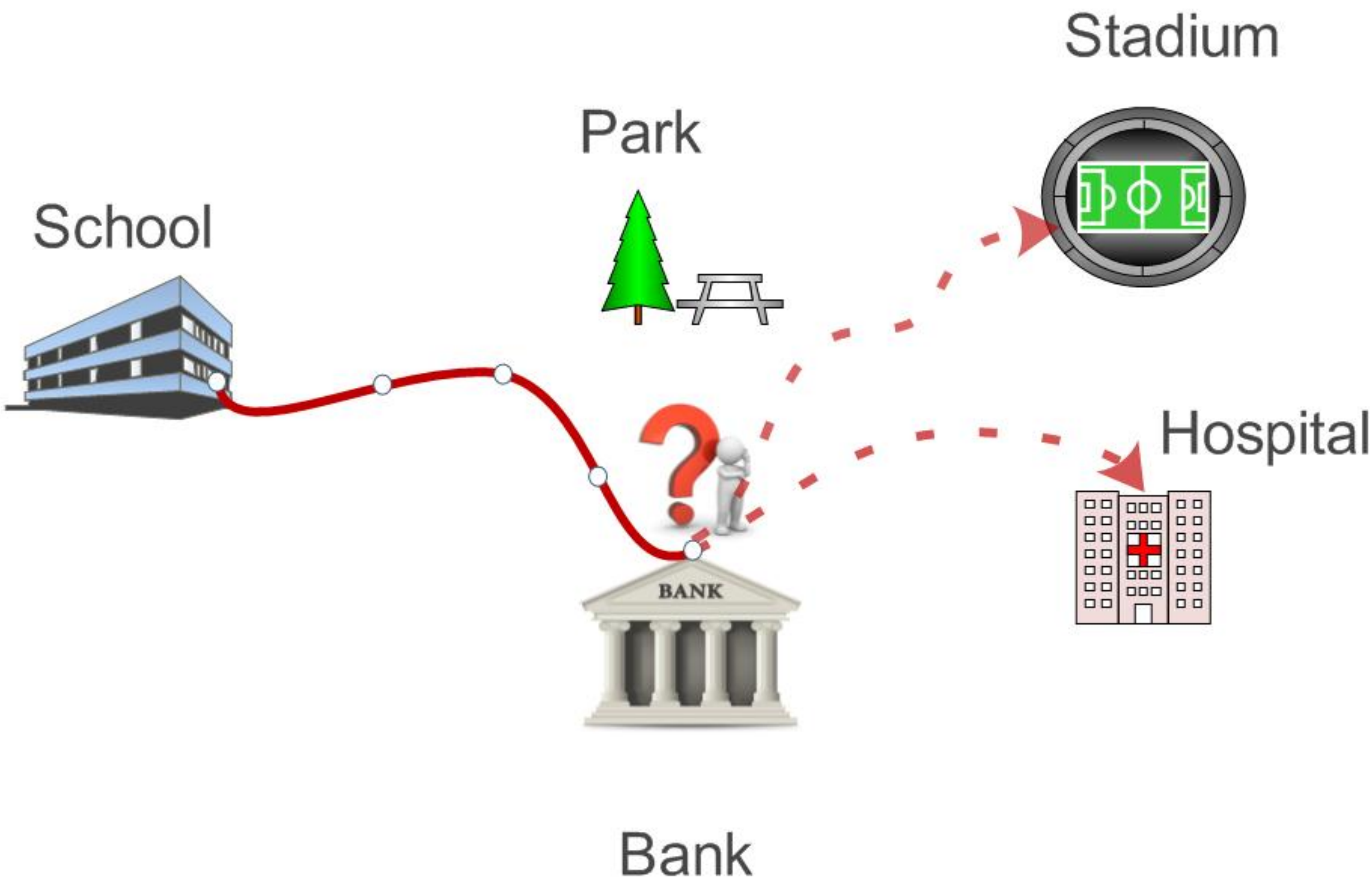
SemanPredict: offline module

- Recap

Input: GPS user's trajectories

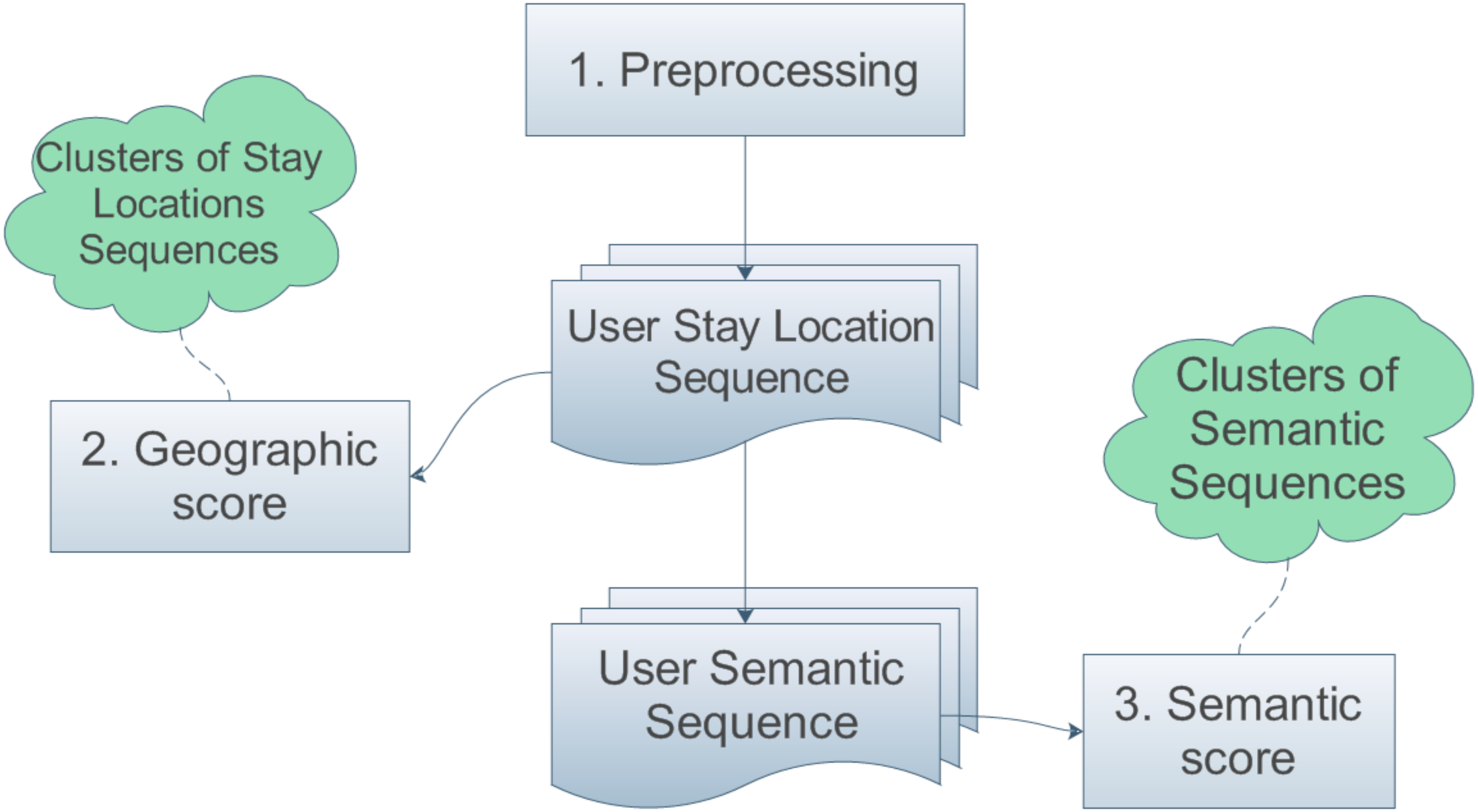


SemanPredict: online module



SemanPredict: online module

- Input: the trajectory of the user's recent moves



SemanPredict: online module

- *Matching score:*

$$Score = \beta \times GeographicScore + \left(1 - \beta\right) \times SemanticScore,$$

$$where\ 0 < \beta \leq 1$$

Note: how well the 'user behavior' matches model

Summarizing Trajectories into k-Primary Corridors: A Summary of Results

Michael R. Evans

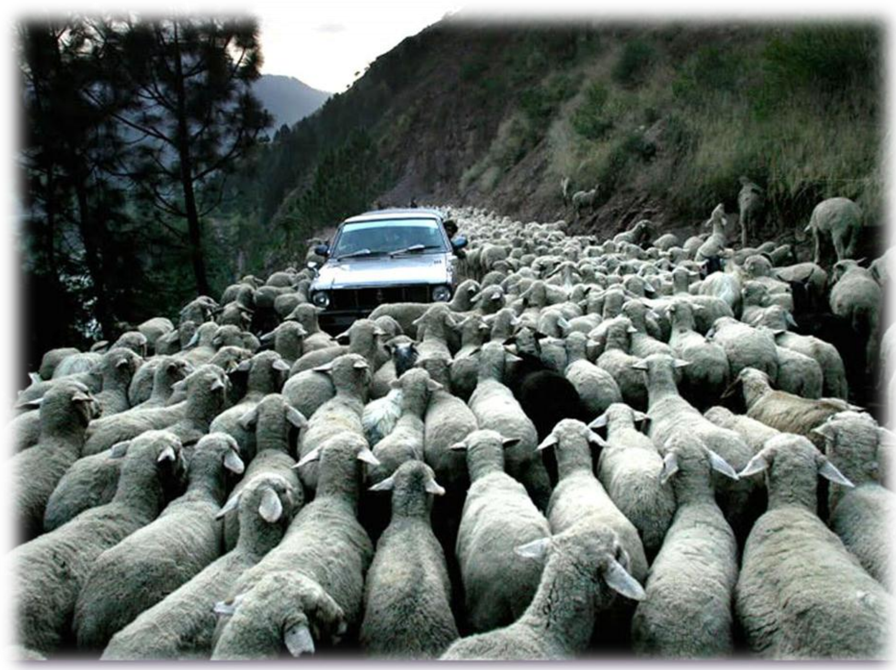
Dev Oliver

Francis Harvey

USA

Why k-Primary Corridors?

- Identifying potential avenues for access

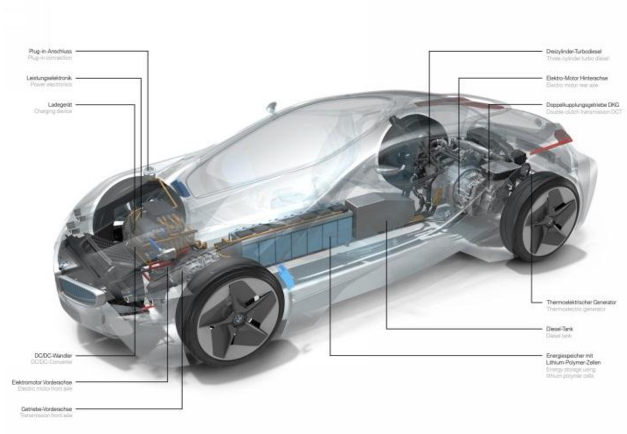


Traffic jam?!!!

- Determine **bus**, **metro** and **railway systems** corridors

Basic concepts

- **Road network**
  - Graph G = {V, E}
  - Nodes = road intersections

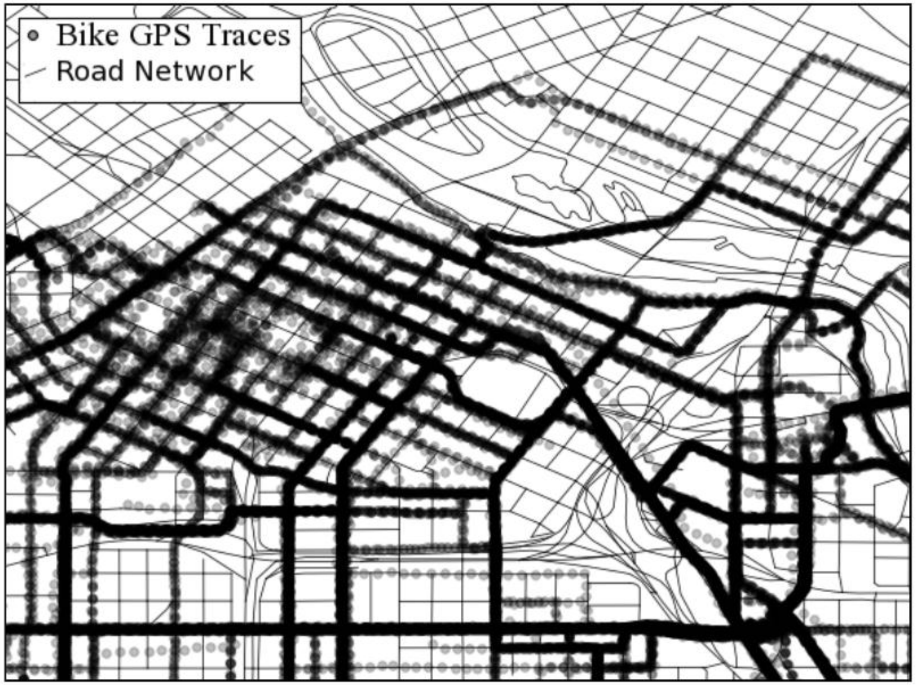


- **Track**
  - GPS trace to series of nodes and edges of road network

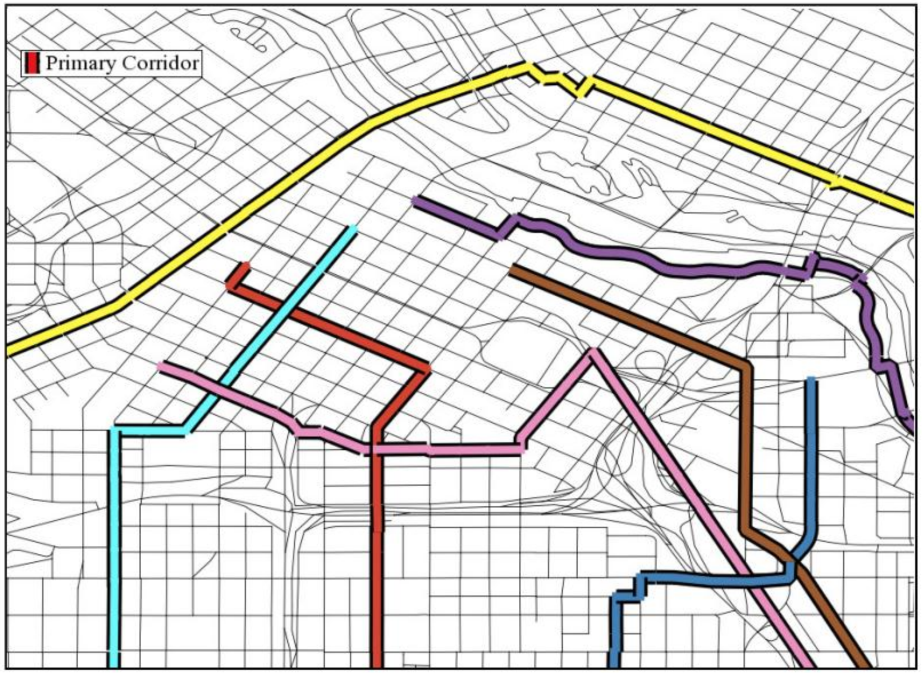
- **Primary corridor**
  - ‘Representative track of group of other tracks’ (Medoid)

- Example:

Road Network



7-Primary Corridors



Problem statement

- **Given**
  - Road Network: G = {V, E}
  - Collection of Tracks: T
  - Number k
- **Find**
  - k-Primary Corridors
- **Constraints**
  - Each **primary corridor is track** itself from set T
  - G is a connected graph with **nonnegative weights**

Solution

- <u>Input:</u> see **Given**
  1. Compute the ***track similarity matrix***

	Track 1	Track 2	Track 3
Track 1	0	1.16	3
Track 2	1.16	0	2
Track 3	3	2	0

  2. Modified k-Medoid clustering [7]

Note: <u>bottleneck</u> is the track similarity matrix computation

- Example: time comparison of steps in k-Primary Corridor problem (100 nodes/tracks, k=5)

Steps of k-PC	Runtime	Percentage
Track Similarity Matrix	46.56 sec	94.1%
Partitioning / Clustering	2.96 sec	5.9%
Total	49.52 sec	100%

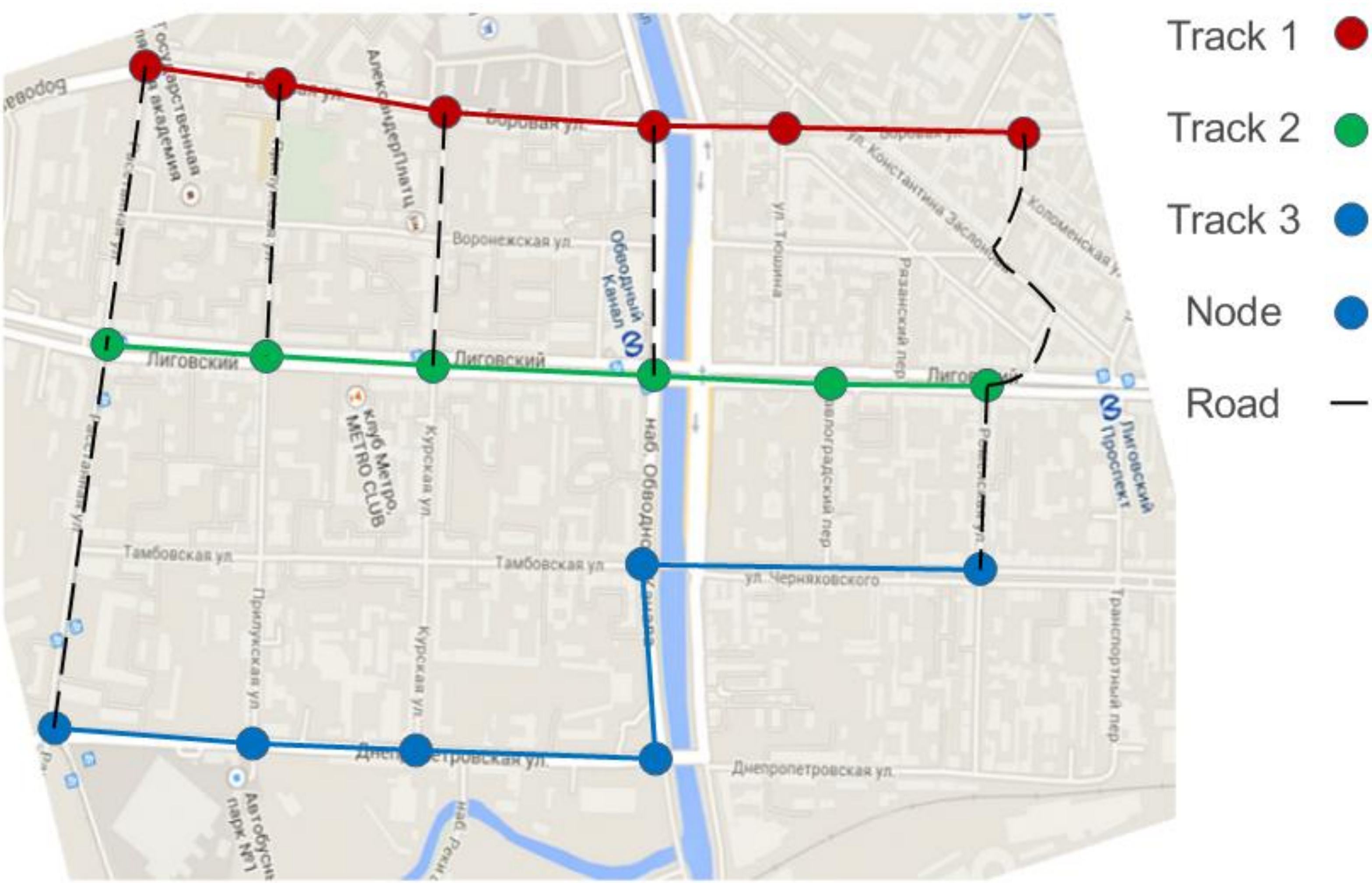
Approaches to compute the Track Similarity Matrix

– ***Graph-Node Track Similarity*** (naïve approach)

1. Consider all track pairs (t_i,t_j)
2. For each pair of nodes in tracks (t_i,t_j) run Dijkstra

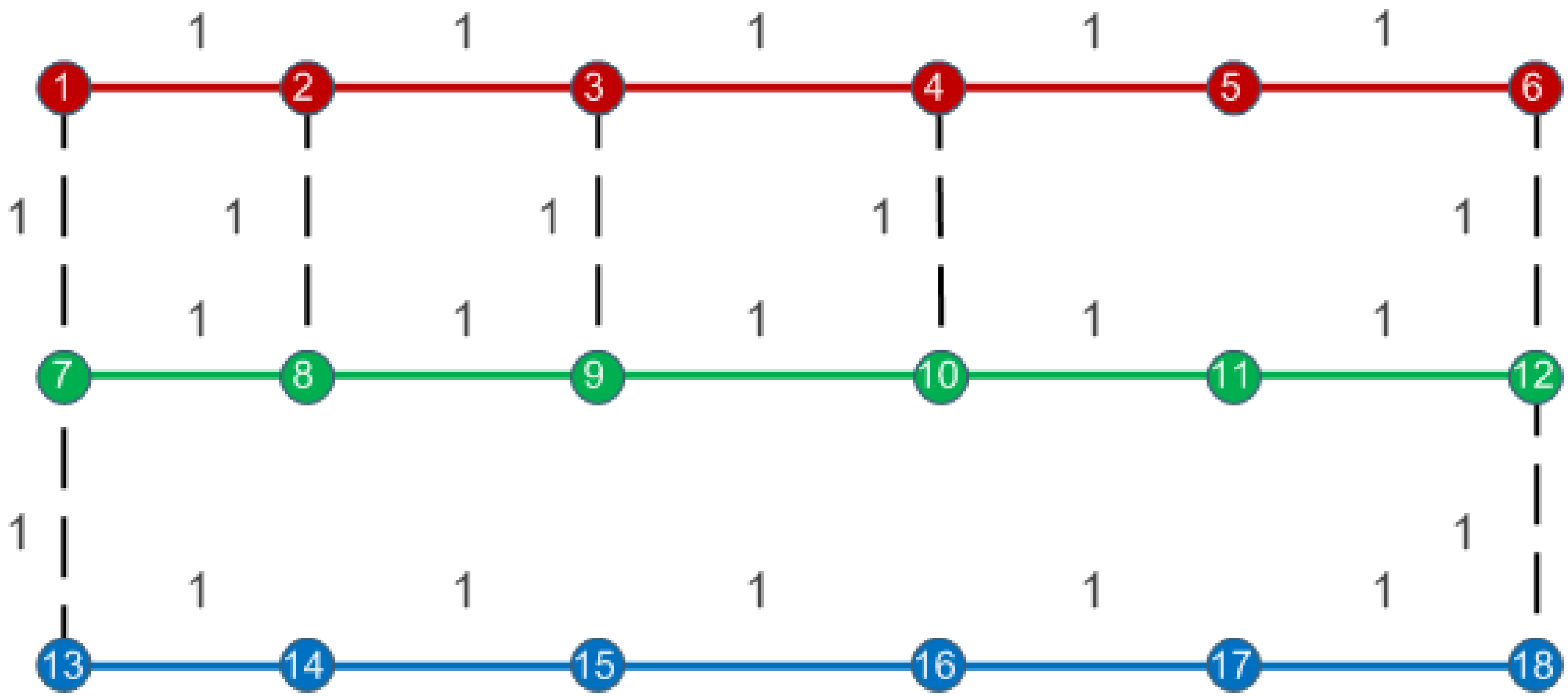
Note: $O(N^2 \cdot N^2 \cdot N \cdot \log N)$

- Example: track similarity matrix computation



- Example: track similarity matrix computation (cont.)

Transform to graph (for simplicity all weights equal 1)



- Example: track similarity matrix computation (cont.)

Calculate **track similarity function**

$$s(t_i, t_j) = \frac{1}{|t_i|} \sum_{n \in t_i} \min_{m \in t_j} \left[ShortestPath(n, m) \right]$$

n, m – nodes of tracks t_i, t_j respectively

	Track 1	Track 2	Track 3
Track 1	0	1.16	3
Track 2	1.16	0	2
Track 3	3	2	0

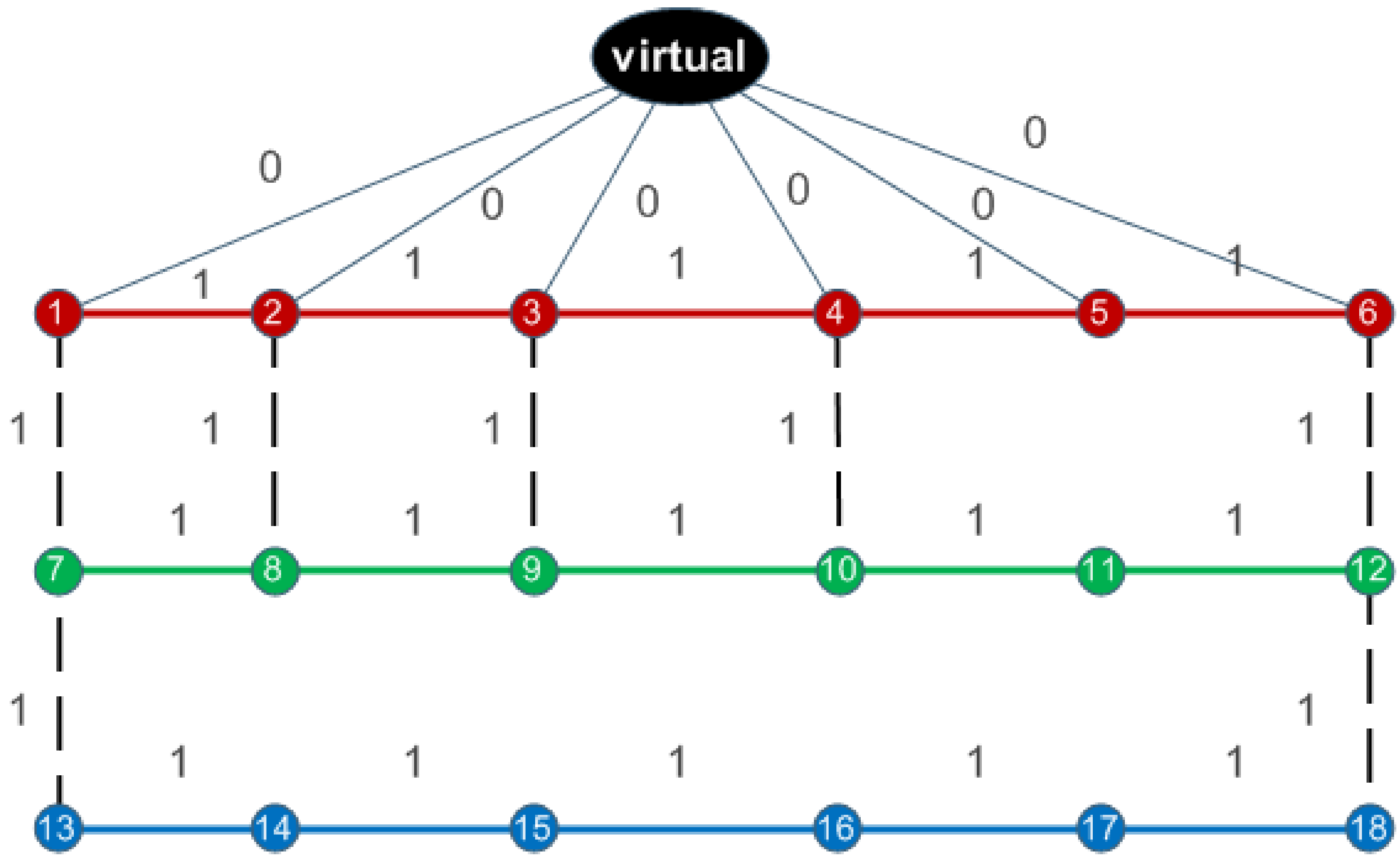
Approaches to compute the Track Similarity Matrix

– **Matrix-Element Track Similarity**

1. Consider all track pairs (i, j)
2. Attach a **virtual node** to track i
3. Run **single** Dijkstra from the virtual node to all the nodes in track j
4. Compute track similarity metric

Note: $O(N^2 \cdot N \cdot \log N)$

- Example: track similarity matrix computation



- Example: track similarity matrix computation (cont.)

Calculate **track similarity function**

$$s\left(t_i, t_j\right) = \frac{1}{\left|t_i\right|} \sum_{m \in t_j} dist[m]$$

$dist[m]$ – distance from node m in tracks $\boldsymbol{t_j}$ to its closest node in track $\boldsymbol{t_i}$

Detecting Road Intersections from GPS Traces

Alireza Fathi

John Krumm

Microsoft Research

USA

2010

Problem definition

1. **GPS data collecting**
   – Expensive:
   Specialized vehicle
   Keep up with changes in the road

   – Chip:
   Regular vehicle
   Detect road intersection

Problem definition

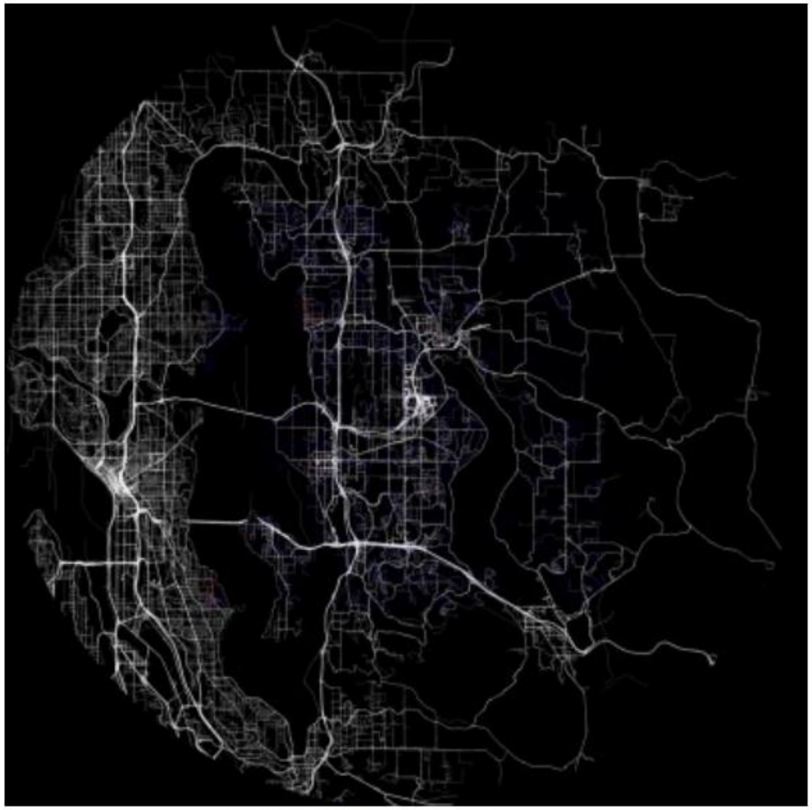
Input: GPS data of regular vehicles

2. Infer the road network from GPS traces:

- Graph G = {V, E}
- Nodes = **road intersections**

Goal: **detect road intersections**

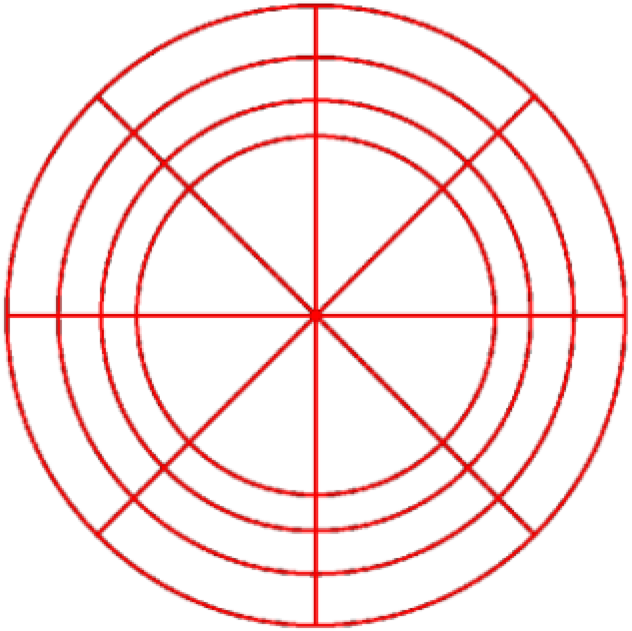
Example: GPS data of regular vehicles

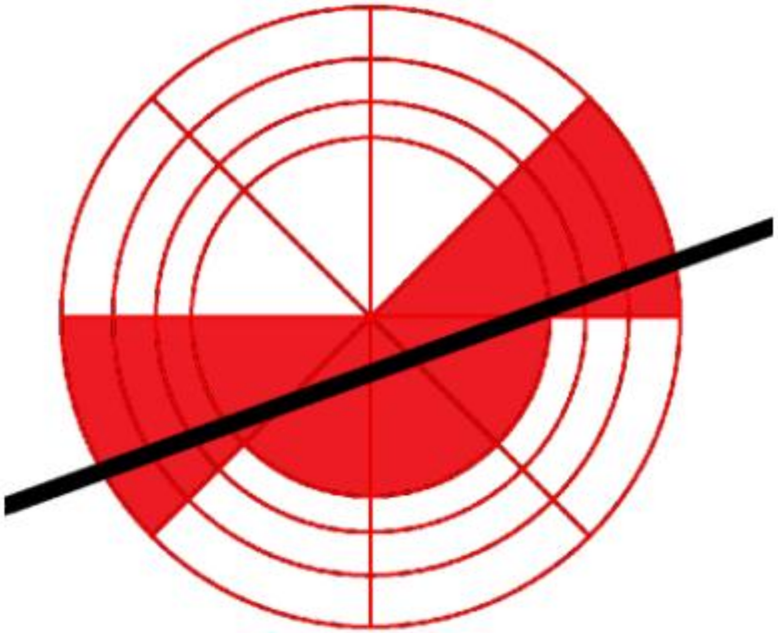


(a) all GPS data

(b) one intersection

Example: Local Shape Descriptor with 32 bins





- Sliding over GPS data to detect intersection

Training phase

Input: examples of GPS trace with intersections and non-intersections

1. Sliding by **shape descriptor** over examples
2. Map the **bins to the feature vector**
3. Learn classifier (Adaboost [8])

Example: result on test dataset

Thank you for your attention!

References

1. Yang Ye*, Yu Zheng, Yukun Chen, Jianhua Feng, Xing Xie. Mining Individual Life Pattern Based on Location History. In proceedings of the International Conference on Mobile Data Management 2009 (MDM 2009). IEEE, 1-10.
2. E. H.-C. Lu and V. S. Tseng. Mining Cluster-Based Mobile Sequential Patterns in Location-Based Service Environments. MDM, 2009
3. A. Monreale, F. Pinelli, R. Trasarti, F. Giannotti. WhereNext: a location predictor on trajectory pattern mining. KDD 2009: 637-646
4. N. Eagle, A. Pentland, and D. Lazer. Inferring Social Network Structure using Mobile Phone Data. In proceedings of the National Academy of Sciences (PNAS),106(36), pp. 15274-15278, 2009
5. Y. Zheng, L. Zhang, and X. Xie. Recommending friends and locations based on individual location history. ACM Transaction on the Web, 2010.
6. J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, V. S. Tseng. Mining User Similarity from Semantic Trajectories. In Proceedings of ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN' 10), San Jose, California, USA, November 2, 2010.

References

7. L. Kaufman, P. Rousseeuw, et al. Finding groups in data: an introduction to cluster analysis, volume 39. Wiley Online Library, 1990.
8. Viola, P. and M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, in Computer Vision and Pattern Recognition (CVPR 2001). 2001. p. I-511- I-518.
9. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.C. Hsu. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proceedings of the 17th International Conference on Data Engineering (ICDE), 2001, 215-224