电子科技大学
University of Electronic Science and Technology of China

Trajectory Pattern Mining on Road Networks

Data Mining Lab,
Big Data Research Center, UESTC
Email：lsl571@163.com

- **Why**

讲这个topic的目的，主要是因为之前讲过的关于轨迹的主题，很少考虑到路网的约束信息。而现实是，我们能得到的轨迹数据集更多情况下都是在路网中的，那么如何在有约束的情况下对具体任务进行模型的建立？所以这次topic就瞄准了路网中的轨迹模式挖掘。

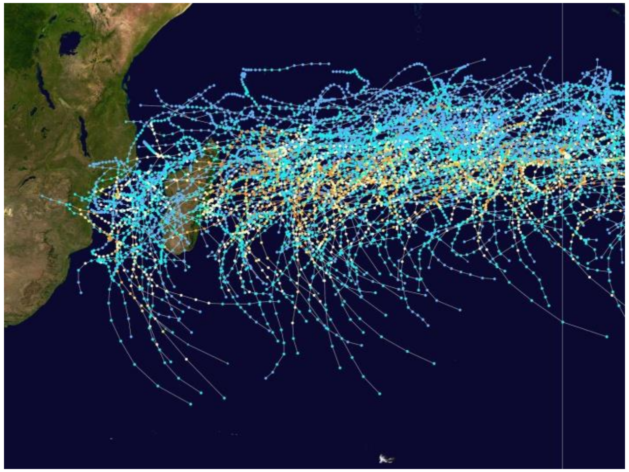
- **Problems**

如果已经考虑到路网存在的约束，我们又该考虑对轨迹添加哪些信息？或者说对于不同的任务，我们考虑的约束信息是否应该不同？等等这些问题都值得讨论和研究。

Plus:在这次准备的过程中，关键是如何讲想表达的主题和要讲到的论文算法结合到一起。经过思考后觉得选择两个相对较为简单的算法，既能将主题表达得比较清楚又能有相对的扩展。

Outline

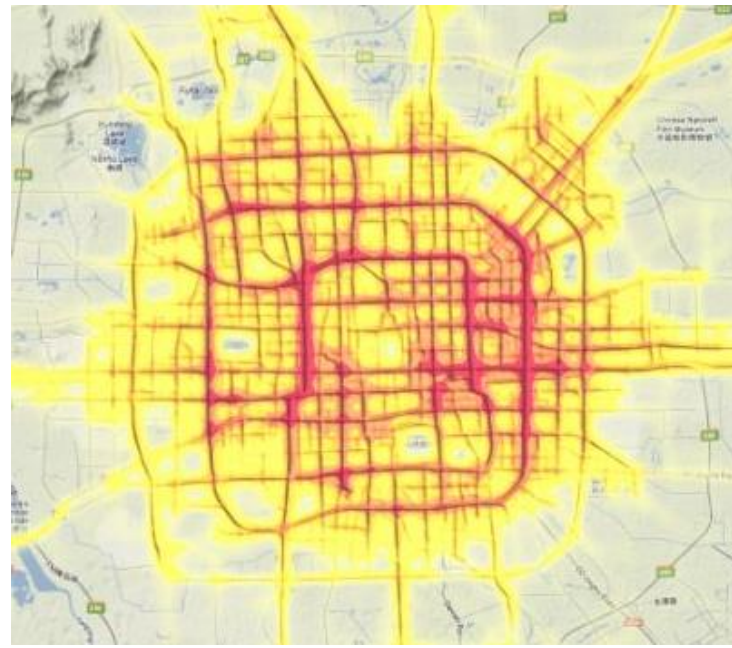
- ✧ **Road Networks & Free Space**
- ✧ **Model Constraint Trajectories**
- ✧ **Problems & Algorithms**



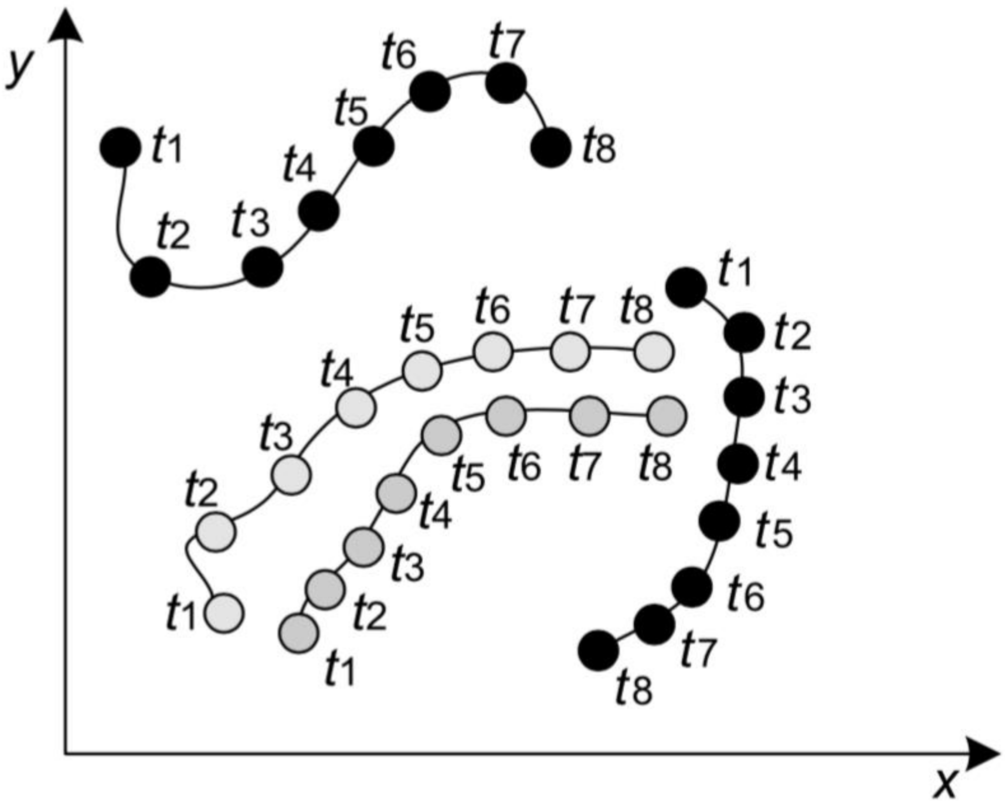
hurricane

v.s.

Animal migration

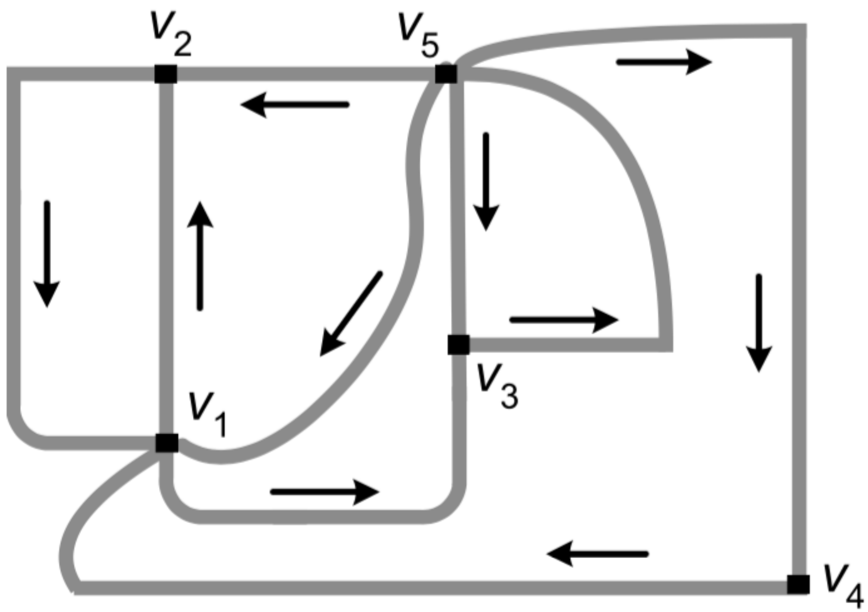


Vehicle density



Example of four trajectories in the 2D Euclidean space

- ✧ **Objects move freely in any direction**
- ✧ **No underlying structure**
- ✧ **Euclidean distance based**

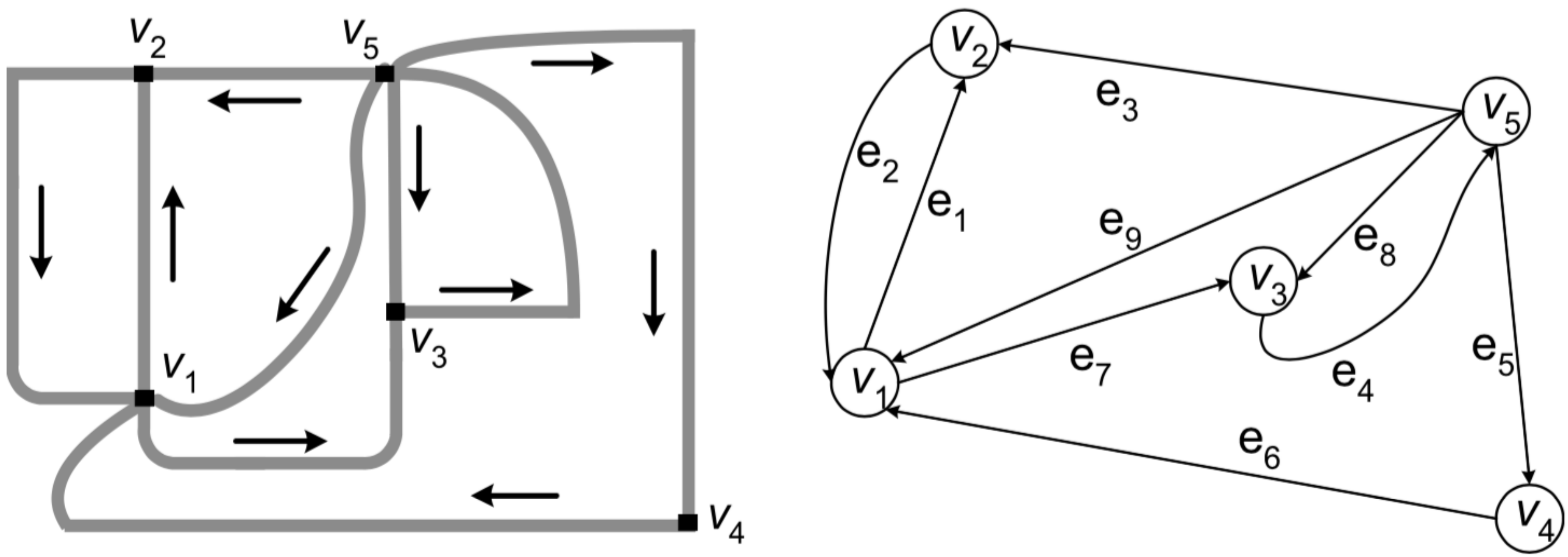


A road network

Some constraints are:

- ✧ **Directions**
- ✧ **Connectivity**
- ✧ **Speed**
- ✧ **Traffic flow**

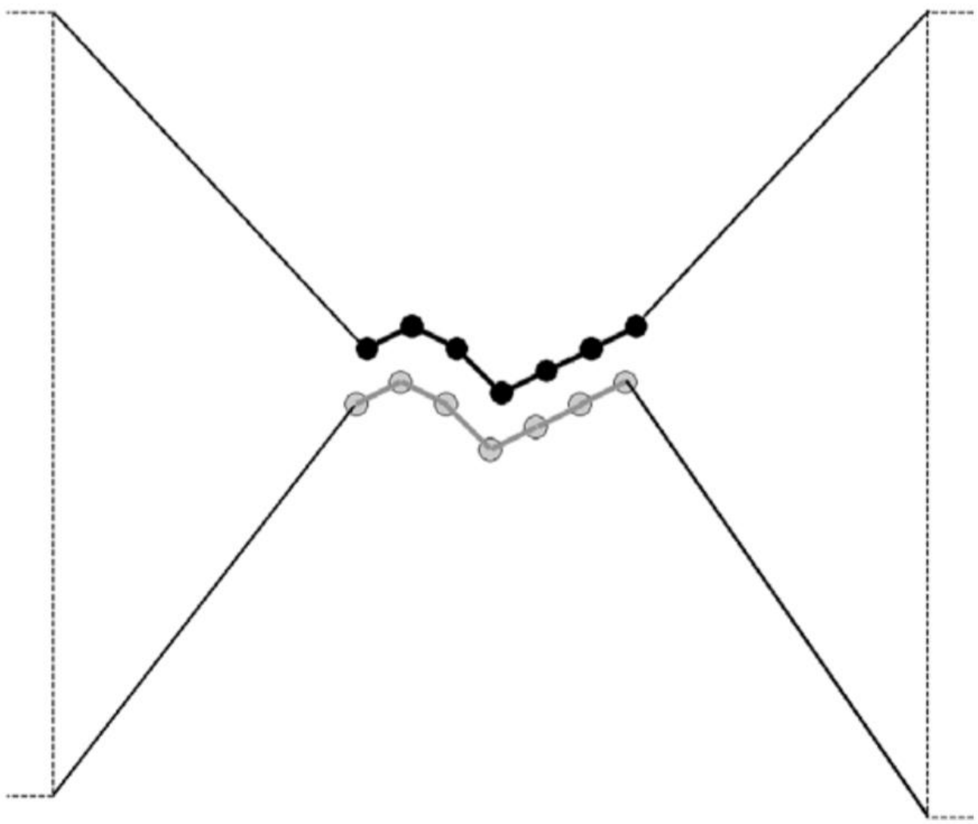
An Natural Transformation → Directed Graph



For every edge,attributes include distance,time interval and other constraints.

Similarity Measure

We can't calculate $d(T_a, T_b)$ according to Euclidean distance.



$$d(v_i, v_j) = \begin{cases} 0, & \text{if } v_i = v_j, \\ \dfrac{c(v_i, v_j) + c(v_j, v_i)}{2D_G}, & \text{otherwise}, \end{cases}$$

$$D_G = \max\{c(v_i, v_j), \forall v_i, v_j \in V(G)\}$$

In a general way, $c(v_i, v_j)$ is defined by Euclidean distance.

$$D_{net}(T_a, T_b) = \frac{1}{m}\sum_{i=1}^{m}(d(v_{ai}, v_{bi})) = \frac{1}{m}\sum_{i=1}^{m}(d(v_{bi}, v_{ai})) = D_{net}(T_b, T_a)$$

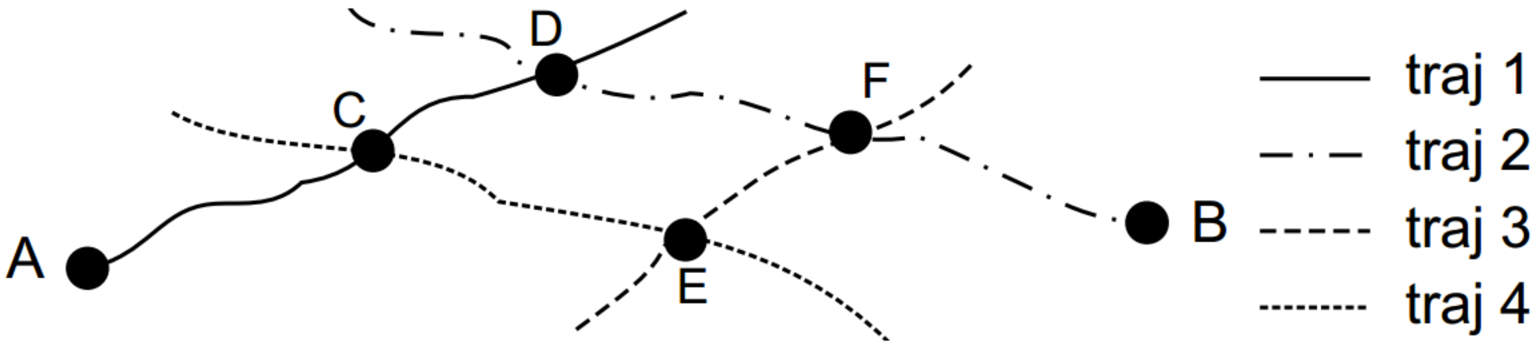
Incorporating Time Information

$$D_{time}(T_a, T_b) = \frac{1}{m-1} \sum_{i=1}^{m-1} \frac{|(T_a[i+1].t - T_a[i].t) - (T_b[i+1].t - T_b[i].t)|}{\max\{(T_a[i+1].t - T_a[i].t), (T_b[i+1].t - T_b[i].t)\}}$$

Finally

$$D_{total}(T_a, T_b) = W_{net} \cdot D_{net}(T_a, T_b) + W_{time} \cdot D_{time}(T_a, T_b)$$

Popular Routes Mining



Objective: A→B If A→C→D is a drive pattern,destination B would be ignored.

$$Pr(n_i \rightarrow n_j) = \frac{number\ of\ trajectories\ on\ (n_i, n_j)}{number\ of\ trajectories\ on\ all\ outgoing\ edges}$$

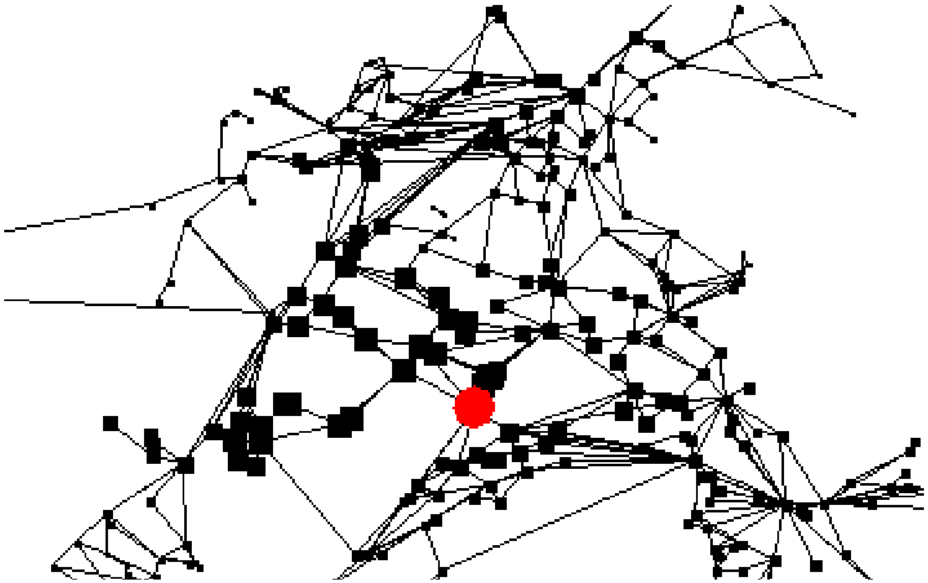
$$Pr_d(n_i \rightarrow n_j) = \frac{\sum_{traj \in (n_i, n_j)} func(traj, d)}{\sum_{traj \in all\ outgoing\ edges} func(traj, d)}$$

$$func(traj, d) = \exp\left(-dist_s(traj, d)\right)$$

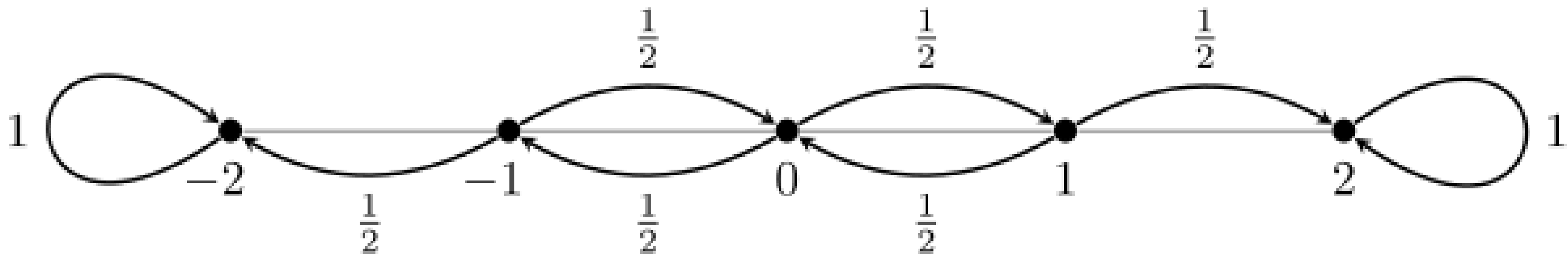
we can consider a travel on such a transfer network based on the turning probability as a **Random Walk** on a directed graph with the transition probability.

$$Pr^t(n_i \to d) = \sum_{j=1}^{t} p^j_{n_i,d}$$

This formula indicates how popular a transfer node ni is, w.r.t. the given destination d.



In the mathematical theory of probability, an **absorbing Markov chain** is a Markov chain in which **every state** can reach an **absorbing state**. An absorbing state is a state that, once entered, cannot be left.



In a road network,destination can be regarded as an absorbing state and each intermediate node is a transient state.

数据挖掘实验室
Data Mining Lab

Absorbing Markov Chain

$$
P = \begin{array}{c|cccc}
 & n_1 & n_2 & \cdots & n_m \\
\hline
n_1 & P(1,1) & P(1,2) & \cdots & P(1,m) \\
n_2 & P(2,1) & P(2,2) & \cdots & P(2,m) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
n_m & P(m,1) & P(m,2) & \cdots & P(m,m)
\end{array}
$$

$$
P(i,j) = \begin{cases} 1 & \text{if } n_i \text{ is an absorbing state } \& \ i = j \\ Pr_d(n_i \to n_j) & \text{if } n_i \text{ is a transient state } \& \ i \neq j \\ 0 & \text{otherwise} \end{cases}
$$

$$p^t_{n_i,d} = \sum_{n_k \in \mathrm{TR}} \left(P^{t-1}(i,k) \cdot P(k,d) \right)$$

$$Pr^t(n_i \to d) = \sum_{j=1}^{t} p^j_{n_i,d}$$

Probability from arbitrary node to destination.

$$Pr^t(n_i \to d) = \textstyle\sum_{j=1}^{t} p^j_{n_i,d}$$

$$= \textstyle\sum_{j=1}^{t} \sum_{n_k \in \mathrm{TR}} \left(P^{j-1}(i,k) \cdot P(k,d) \right)$$

$$n_i.popularity(d) = Pr^t(n_i \rightarrow d)$$

$$\rho(R) = \prod_{j=1}^{i} n_j.popularity(d)$$

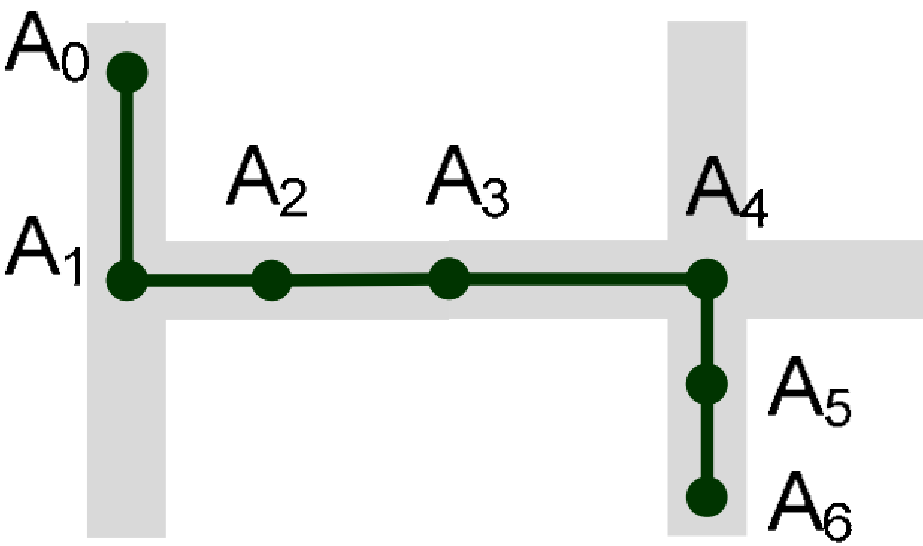
Maximizing this formula ,we can get one popular route.

Road Network Aware Trajectory Clustering

In the context of a road network, those objects have similar movement behavior with respect to the road segment.

Atomic object: Road segment

Network proximity + Traffic flow + Speed limit + …



$$TR = \{trid, l_0 l_1 \ldots l_n\}$$

$$tf = \{trid, sid, l_k l_{k+m}\}$$

A trajectory has three t-fragments.

Base clusters: $S = \{tf_i | TR(tf_i) \in \dot{\mathcal{T}}, tf_i.sid = e.sid\}$

Dense-core: $S = \arg\max_{S_i \in B} |S_i|$

Netflow: $f(S_i, S_j) = |PTr(S_i) \cap PTr(S_j)|$

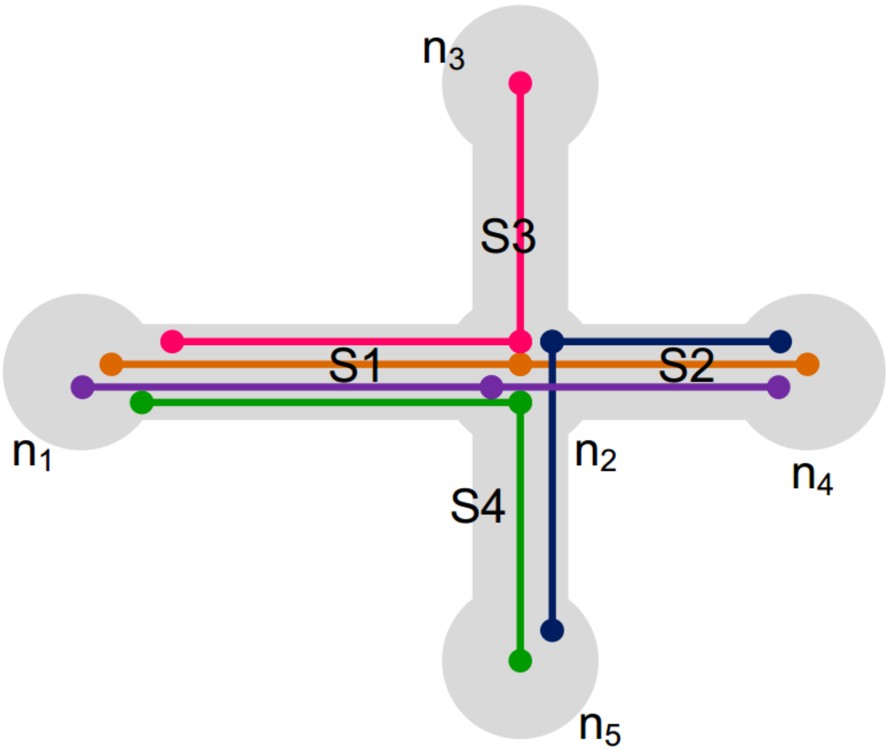
F-neighbor: $N_f(S_i, n_u) = \{S_j \,|\, e^{S_j} \in L_{n_u}(e^{S_i}) \,\&\, f(S_i, S_j) > 0\}$

maxFlow-neighbor:

Flow cluster:

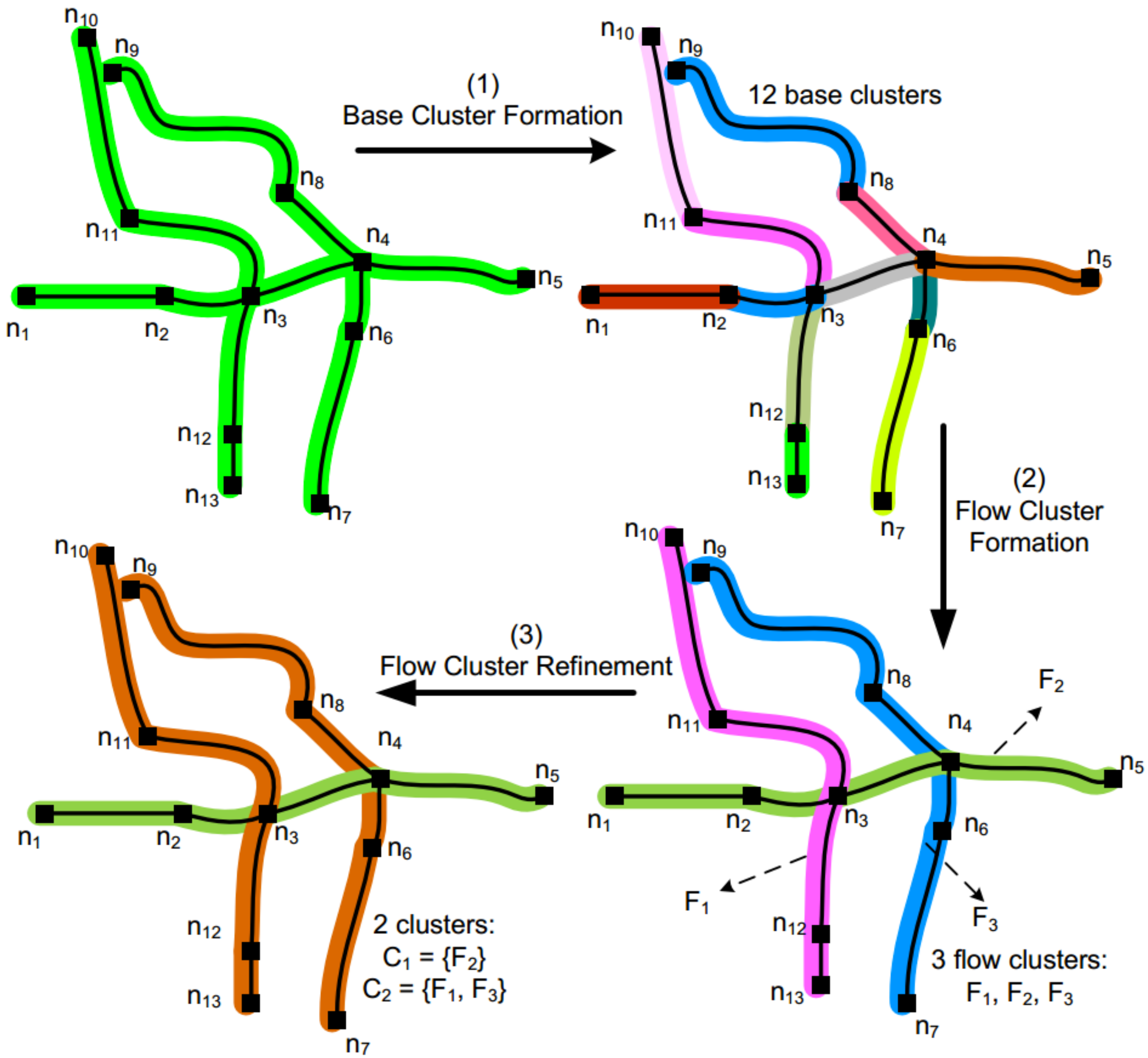
$$F = \{S_0, S_1, \ldots, S_N\}, \text{ where } S_{i+1} \in N_f(S_i)(0 \leq i < N)$$

$$f(S_1, S_2) = 2,\ f(S_1, S_3) = 1\ldots$$



An example of base clusters and flow cluster

Three phases of this algorithm



1. Base Cluster Formation
-- Skip

2. Flow Cluster Formation:

Density-based Flow Cluster Initialization: Dense-core of the set of base clusters.

Merging criteria: Flow + Density + Speed Limit

Flow factor: $$q = \frac{f\left(S, S_j\right)}{|PTr(S)|}$$

Density factor: $$k = \frac{d\left(S_j\right)}{d(S) + \sum_{S_i \in N_f(S, n_u)} d\left(S_i\right)}$$

Speed limit factor: $$v = \frac{speed\left(S_j\right)}{\sum_{S_i \in N_f(S, n_u)} speed\left(S_i\right)}$$

$$SF = w_q \times q + w_k \times k + w_v \times v$$

How to set ?

Which flow cluster should a base cluster belong to?

$$f_{n2n} / f_{c2n} > \beta \ ?$$

3. Flow Cluster Refinement

Hausdorff distance + DBScan

Reference

数据挖掘实验室
Data Mining Lab

1.Chen Z, Shen H T, Zhou X. Discovering popular routes from trajectories[C] //Data Engineering (ICDE), 2011 IEEE 27th International Conference on. IEEE, 2011: 900-911

2.Lee J G, Han J, Li X, et al. Mining discriminative patterns for classifying trajectories on road networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(5): 713-726.

3.Lee J G, Han J, Li X, et al. Mining discriminative patterns for classifying trajectories on road networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(5): 713-726.

4.Han B, Liu L, Omiecinski E. Neat: Road network aware trajectory clustering[C] //Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on. IEEE, 2012: 142-151.

5.Wu H, ZHENG B. Probabilistic Robust Route Recovery with Spatio-Temporal Dynamics[J]. 2016.

6.Li X, Han J, Lee J G, et al. Traffic density-based discovery of hot routes in road networks[C]//International Symposium on Spatial and Temporal Databases. Springer Berlin Heidelberg, 2007: 441-459.

Thanks