Inferring Traffic Cascading Patterns

Yuxuan Liang[1,2], Zhongyuan Jiang[1], Yu Zheng[2,1,3]

[1] Xidian University

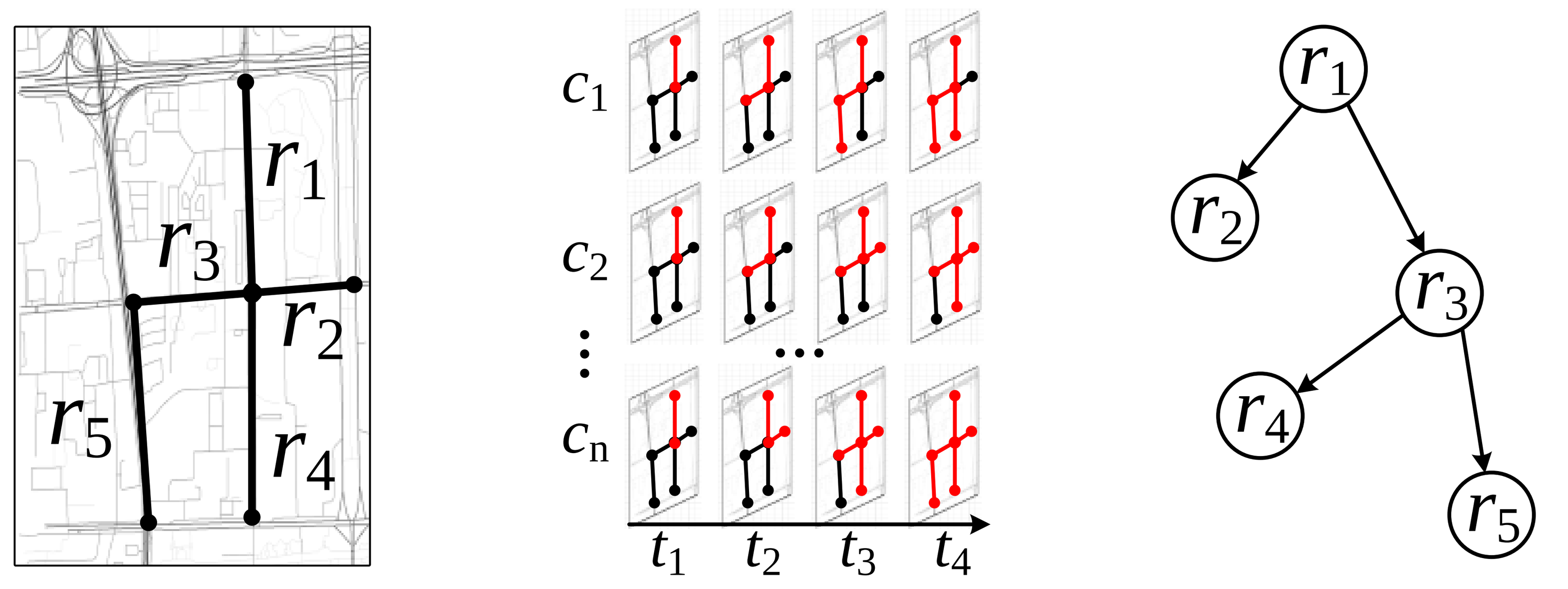
[2] Microsoft Research, Beijing, China

[3] Chinese Academy of Scienses

Microsoft Research 微软亚洲研究院

Released Codes & Paper

Introduction



(a) Road Network (b) Observed Cascades (c) Cascading Pattern

Knowing the traffic cascading pattern can help

- predict future traffic conditions
- identify bottlenecks of road networks

Challenges

- Implicit interaction
- Multiple sources
- Geospatial correlation

Contribution

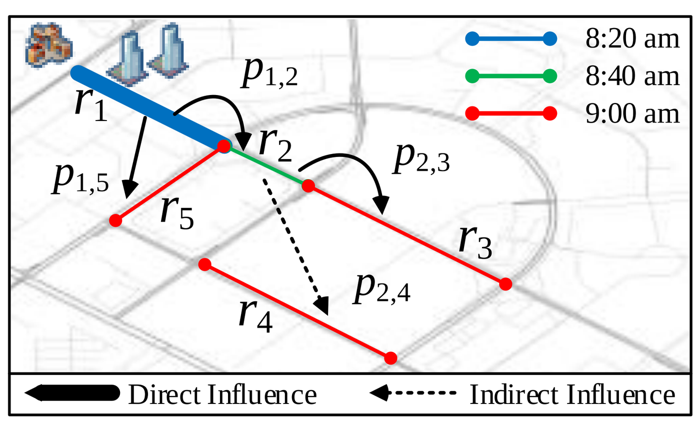
- Modeling three-fold influence
- **Cascading pattern inference**
- Real evaluation

Insight

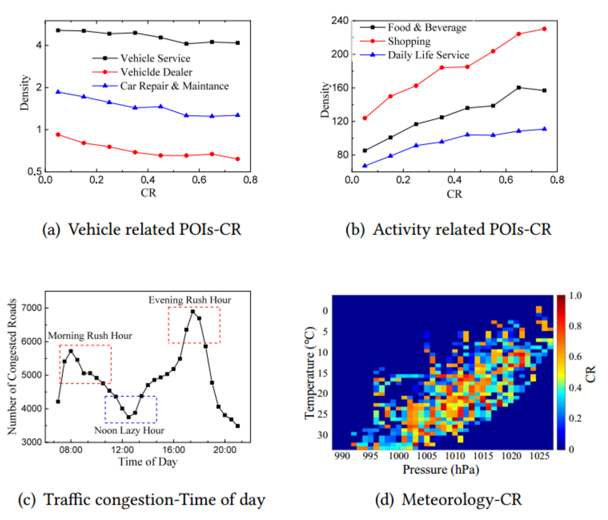
Three-fold influences

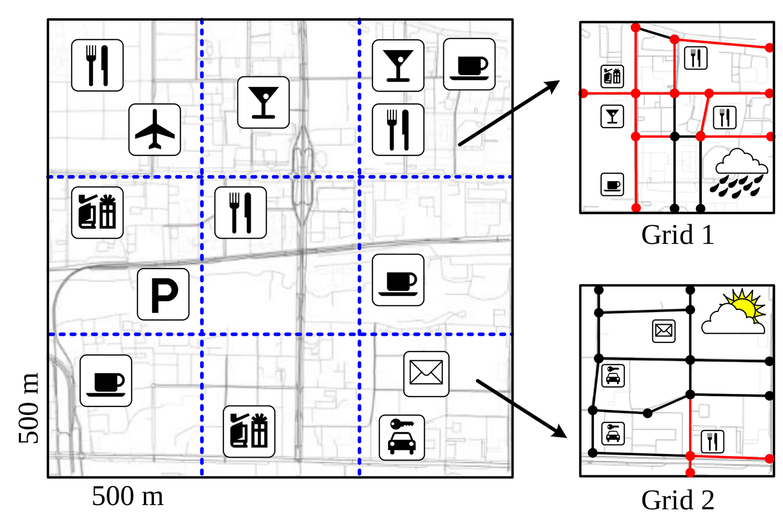
- Direct influence
- Indirect influence

$$f(t_i|t_j; a_{j,i}, \lambda) \propto e^{-a_{j,i}(\Delta_{j,i}+\lambda * d_{i,j})}$$

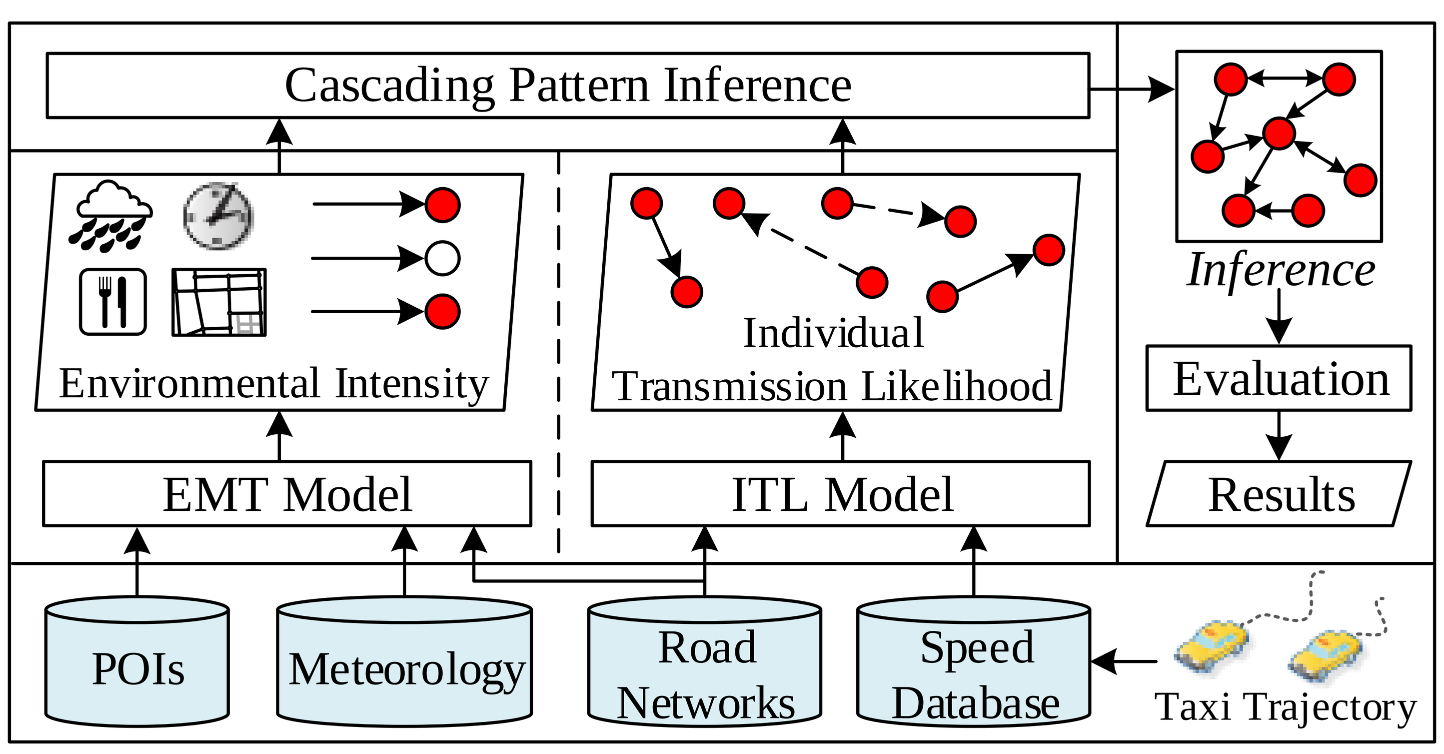


- Environmental influence





Overview



Methodology

Cascading Pattern Construction

- Given a propagation tree T, the likelihood of a cascade c:
- Given a network G considering all possible trees, the likelihood of a cascade c:
- Conditional independence assumption

$$f(c|T) = \prod_{(j,i)\in E_T} f(t_i|t_j; \alpha, \lambda),$$

$$f(c|G) = \sum_{T\in\mathcal{T}_c(G)} f(c|T)P(T|G),$$

$$f(C|G) = \prod_{c\in C} f(c|G).$$

Network Inference

$$\hat{G} = \arg\max_{|G|\le k} f(C|G),$$

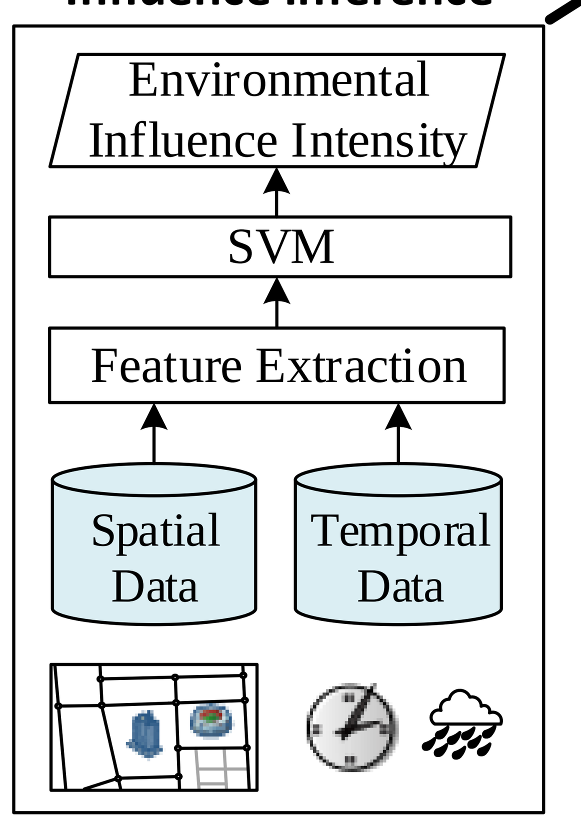
Alternative Target

$$\hat{G} = \arg\max_{|G|\le k} F(C|G)$$

$$F(c|G) = \sum_{t_i\in c} log \sum_{t_i\in c: t_j\le t_i} w_c(j,i),$$

$$w_c(j,i) = \theta\varepsilon_g^{-1} f(t_i|t_j; a_{j,i}, \lambda)$$

Environmental Influence Inference



Approximate Algorithm

Algorithm 1 Approximate algorithm for *CasInf*

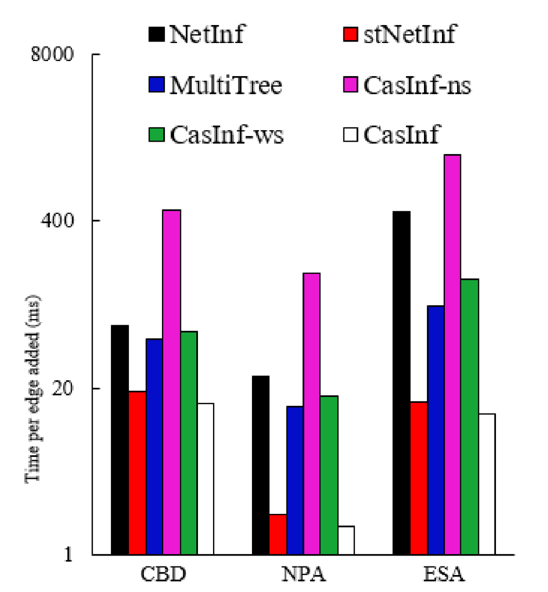
Input:
k : the number of edges in cascading patterns we infer.
C : the set of cascades obtained in a time span.
$\mathcal{D}$: a constant denoting the spatial constraint.

Output: G: the inferred cascading pattern.

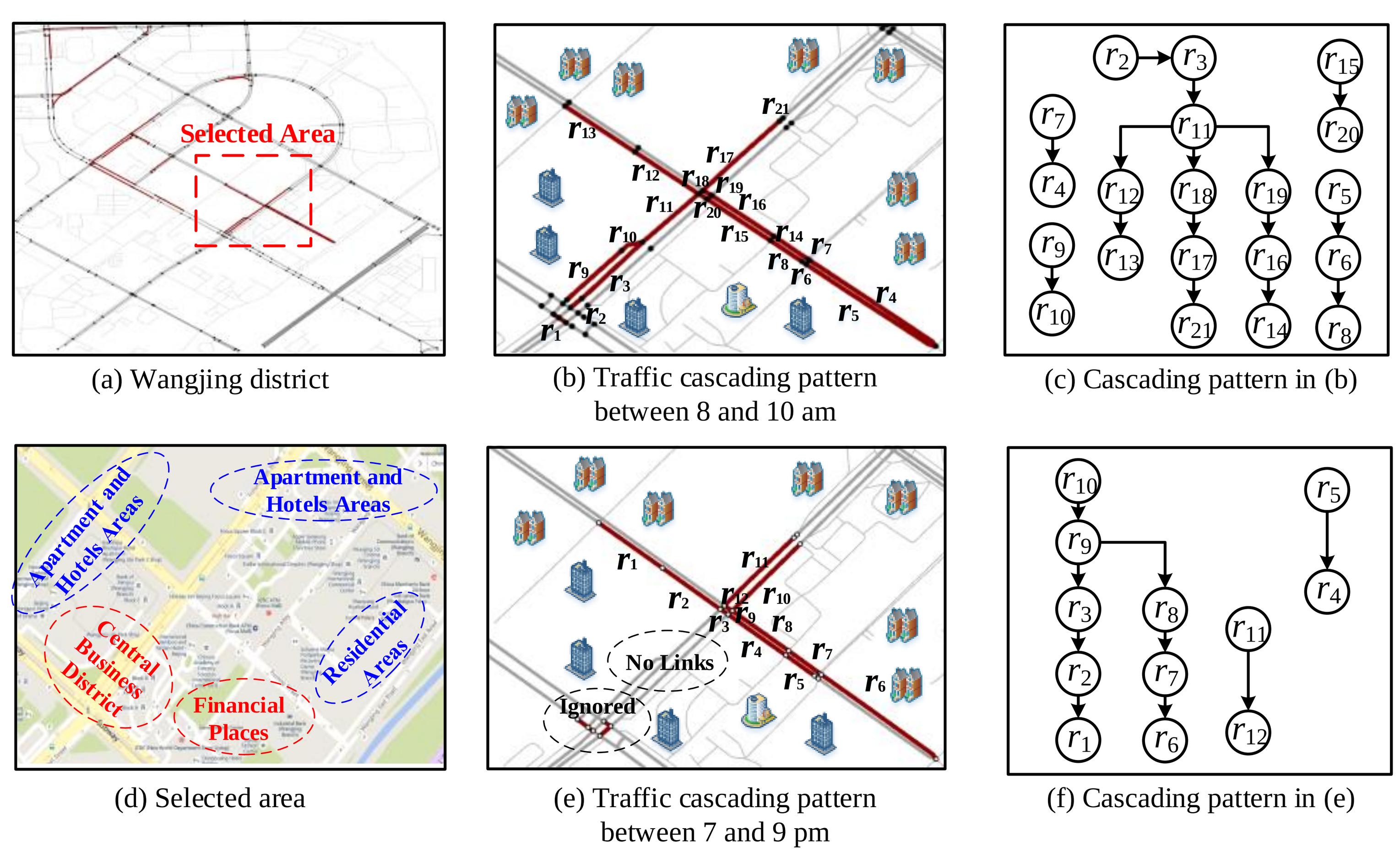
```
1: G ← K̄;
2: P ← all pairs (j, i): ∃c ∈ C with t_j < t_i and d_{j,i} < 𝒟
3: while |G| ≤ k do
4:   for all (j, i) ∈ P\G do
5:     δ_{j,i} = 0;
6:     for all c : t_j < t_i do
7:       ε_g ← environmental intensity inference
8:       w_c(m, n) ← weight of (m, n) in G ∪ (j, i);
9:       for all m : t_m < t_i and m ≠ j do
10:        δ_{c,j,i} = δ_{c,j,i} + w_c(m, i);
11:      end for
12:      δ_{j,i} = δ_{j,i} + log((δ_{c,j,i} + w_c(j, i)) / δ_{c,j,i});
13:    end for
14:  end for
15:  (j*, i*) ← argmax_{(j,i)∉G} δ_{j,i};
16:  G ← G ∪ (j*, i*);
17: end while
```

Evaluation

Methods	CBD	NPA	ESA	Overall
NetInf	0.270	0.119	0.116	0.168
stNetInf	0.308	0.201	0.394	0.301
MultiTree	0.311	0.140	0.141	0.197
FBM	0.287	0.193	0.171	0.217
STC-DBN	0.307	0.198	0.225	0.243
CasInf-gd	0.359	0.258	0.488	0.368
CasInf-td	0.336	0.199	0.203	0.246
CasInf-ne	0.363	0.298	0.515	0.392
CasInf-ni	0.197	0.129	0.215	0.180
CasInf	**0.384**	**0.317**	**0.545**	**0.415**



Case Study



(a) Wangjing district (b) Traffic cascading pattern between 8 and 10 am (c) Cascading pattern in (b)

(d) Selected area (e) Traffic cascading pattern between 7 and 9 pm (f) Cascading pattern in (e)