Optimal Seeds Discovery of Traffic Congestions

Carlos Bermejo*, Ting Wu*, Xiang Su†‡, Pan Hui*†
*Department of Computer Science and Engineering, Hong Kong University of Science and Technology
†Department of Computer Science, University of Helsinki ‡Department of Computer Science, University of Oulu
cbf@cse.ust.hk, twuad@cse.ust.hk, xiang.su@helsinki.fi, panhui@cse.ust.hk

***Abstract*—With the rapid adoption of wireless sensor networks (WSNs) into smart cities and vehicle networks, traffic problems can be evaluated and predicted in real-time. In this paper, we propose a data-driven approach to find out the most influential causes of traffic congestions. We first find the top most influential regions and use the *Fortune's* algorithm to partition the city. Second, we propose a model with three correlations to measure the dependency between two traffic events, which are spatial correlation, temporal correlation, and logical correlation. Third, we adapt the *Independent Cascade model* with a pruning algorithm to address traffic congestions. At last, we conduct intensive experiments on large real-world GPS trajectories generated by more than 10,200 taxis in Shanghai to demonstrate the performance of our approaches.**

I. INTRODUCTION

Traffic congestions are one of the most crucial problems in major cities in the last several decades. The future adoption of WSNs in vehicles and urban environments can improve our understanding of basic laws governing the mobility patterns of vehicles. Smart cities will open new possibilities to achieve real-time congestion predictions [1] and a more refined collection of data. The mobility pattern is regarded as an effective angle to inspect the traffic condition at the city scale [2], [3], making its characterization and modeling an active topic in traffic anomaly study. In particular, the GPS tracking data generated by automobiles provides us with great opportunities to infer rich knowledge, which can be leveraged in the improvement of thoroughfares layout and the optimization of transportation infrastructure. Previous works have shed light on analyzing the location traces of moving objects, such as finding the "periodic" or frequent movement patterns [4] as well as abnormal movement patterns [5].

Aiming at triggering the most influential change, we ask, can we *discover those areas which have the most significant influence on the urban traffic condition as a whole to make the best use of limited resources?* Such optimizations become a primitive for urban computing applications, guiding thoroughfares layout design, infrastructure reconstruction, and instant congestion resolution. We name it as *seeds discovery* and will investigate its real-life feasibility in this paper.

In this paper: differing from the major cause discovery [6], [7], the core idea of *seeds discovery* is introducing a problem to find the top J most influential regions (J-KEY-Region) to a graph representing congestion network and analyzing its diffusion process to address this problem. The following challenges should be addressed in order to have a feasible system: *(1)* detecting traffic congestion from vehicles' trajectories data; *(2)* quantifying the dependence among different congestion events; and *(3)* modeling the diffusion process. Since traffic congestions are treated as anomalies in this topic, and there is already much work on anomalies and outliers detection [5], we choose a different approach. Inspired by [8], we adapt the *Independent Cascade model* to address traffic congestions.

Summary of contributions:

- We identify *seeds discovery* problem as a *J-KEY-Region* problem to find the top J most influential regions.
- We introduce a novel way to partition the city map by adopting *Fortune's algorithm* [9] to get the *Voronoi diagram*.
- We propose an integrated correlation rule to quantify the dependency between congestion events.
- We adapt *Independent Cascade* model [10], [11] to address the *J-KEY-Region* problem and design a pruning strategy to improve the efficiency.
- We experiment on a large real-dataset containing more than 10,000 taxicabs in Shanghai and show the effectiveness and efficiency of our approaches.

II. MODELS AND PROBLEM DEFINITIONS

Now we define the terminologies used throughout the paper and give an overview of our approach, see Table I.

A. *Traffic Congestion Detection*

Traffic congestion is formally defined as a condition on road networks that occurs as usage increases and is characterized by slower speeds, longer trip times, and increased vehicular queuing [12]. We detect traffic congestions analyzing the traffic speed of the vehicles.

For the sake of generality, we choose the region as the basic spatial unit in our model. The comparison of the *Region-based model* and *Road-based model* will be discussed in more detail in Section IV. Considering that state-of-the-art models partition the city according to major road [13], which suffers from a fundamental *boundary* issue [6], we propose a novel approach to derive new regions based on that. In the same way, as described in [13], we partition a city into disjointed regions using the major roads of a city. *Connected Components Labeling* is employed to partition effectively and efficiently [14]. Then, we get centroids of all regions. Afterward, we get a set of N points. Here for the concern of

TABLE I
MEANINGS OF SYMBOLS USED

Notation	Description
F_i	Boolean indicator shows whether traffic congestions occur or not
R	set of regions
E_i/ $E_i.r_i$	traffic congestion in a patio-temporal unit
$TC(E_i, E_j)$	time correlation between two events
$SC(E_i, E_j)$	spatial correlation between two events
$LC(E_i, E_j)$	logical correlation between two events
$p_{u,v}$/ $p(u \rightarrow v)$	influence probability from node u to v in independent cascade model
$\sigma(.)$	the expectation of total weight of events influenced by selected region
Φ	one of the possible outcomes of the traffic events network
Ψ	partial realization

simplicity, we assign points with the average value of highest and lowest longitude and latitude.

$$P_i = (\frac{max\{X_i\} + min\{X_i\}}{2}, \frac{max\{Y_i\} + min\{Y_i\}}{2}), \quad (1)$$

where X_i is the set of all possible *longitude* in the *Region i*, Y_i is the set for *latitude*.

We adopt *Fortune's algorithm* [9] to generate a *Voronoi diagram* with the city map and the set of points P. After this process, every point on the map is in the region whose centroid is the closest to it. Thus in the subsequent modeling, the bias in the spatial correlation from various scales of original regions declines. Again, for computational simplicity, we use *Manhattan Distance* in *Fortune's algorithm*. For each region, we partition the spatial region into T spatial, temporal units. We detect the traffic congestions in each Spatio-temporal unit and mark those units with traffic congestions as infected units. Each infected unit represents a traffic event E_i, and continuous infected units in the same region will be merged as a single traffic event. As we are only interested in the influences between different traffic congestions, we do not parametrize the duration of given congestion. The concrete steps are:

- Computing the average speed $v_{normal}.r_i$ of each region for each day. We use it as the normal speed of this region.
- Computing the average speed $v_{ins}.r_i$ of each region for each time interval. We treat V_i as the instantaneous speed of each Spatio-temporal unit.
- Comparing every $v_{ins}.r_i$ with $v_{normal}.r_i$. We identify Spatio-temporal units with congestions by:

  $$F_i = \begin{cases} 1 & if\ v_{ins}.r_i \leq \alpha v_{normal}.r_i \\ 0 & otherwise \end{cases} \quad (2)$$

  where F is a Boolean indicator shows whether traffic congestions occur or not, and α is between 0 and 1.

By adopting the approach above, we identify the Spatio-temporal units that encounter traffic congestions. In other words, we detect traffic congestions in specific regions within certain periods.

B. Influence Measurement

We dilate how to compute the influence probability in the network. If the traffic event E_i happens, the traffic event E_j will happen with the probability $P_{ij} (0 \leq P_{ij} \leq 1)$, we add a directed edge from E_i to E_j with weight P_{ij}.

Definition 1 (Influence): The influence from event E_i to E_j is defined as the probability $P_{ij} (0 \leq P_{ij} \leq 1)$ when E_i happens then E_j also happens afterwards.

The principle underlies this probability is the same as that underlies correlation, which is specified in the covariance matrix L consisting of average passing time in the basic spatial unit. In [8], authors captured correlation by appropriately specifying the structure of the covariance matrix L, consisting of average passing time in the basic spatial unit. The spatial correlation between route links is captured by LL^T, while temporal correlation is captured by $L^T L$, [15] calculates these correlations in a more sophisticated fashion. Covariance is calculated from traffic data that have been calibrated based on the Stochastic Cell Transition Model to represent the correlation. However, both [8] and [15] build their model on low-level data, taking input directly from traffic data like traffic flows or passing the time. From their points of view, correlation mining is a means to decipher current traffic conditions. While in our work, traffic anomalies detected from the previous phase are objects, we intend to measure correlations, which usually have attributes like location and time that are conventionally utilized in correlation measurement. Hence though well elaborated, these state-of-the-art models do not match our requirements. We propose an integrated correlation to measure the influence probability between two traffic events by combining spatial correlation, temporal correlation, and logical correlation (all the correlation values are normalized in the domain of $[0, 1]$).

1) Temporal Correlation: The temporal correlation $TC(E_i, E_j)$ between traffic event E_i and traffic event E_j is defined as the confidence of the association rule $E_i \rightarrow E_j$ under the temporal constraint that traffic event A is before traffic event E_j, and their difference does not exceed a threshold h_t. When two events satisfy the time constraint, they can be counted as one support for the rule $E_i \rightarrow E_j$.

2) Spatial Correlation: Recall that the map is partitioned into N regions. The spatial correlation of two traffic events is computed as the inverse of Euclidean distance between the geographical centroids of two traffic events (SC).

$$SC(E_i, E_j) = \frac{1}{\sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)}} \quad (3)$$

where (x_i, y_i) and (x_j, y_j) are the geographical centroids of event E_i and E_j respectively.

Recall that the map is partitioned into N regions. The spatial correlation of two traffic events is computed as the inverse of Euclidean distance between the geographical centroids of two traffic events:

$$SC(E_i, E_j) = \frac{1}{\sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)}} \quad (4)$$

where (x_i, y_i) and (x_j, y_j) are the geographical centroids of event E_i and E_j respectively.

3) Logic Correlation: The conventional measurement of correlation may engender discrepancy in the situation that two viaducts leading to different destinations may be adjacent to each other at some points, where congestion at one viaduct will little likely affect the other one. We filter these errant factors out using the inner relationships of traffic flows that are scrutinized through logical correlation mining. In our work, logical correlation is defined as the volume of identical vehicles' appearing in different events, as shown in the following Chi-Square formula

$$LC(E_i, E_j) = \frac{(N_{11}^{ij}N_{00}^{ij} - N_{10}^{ij}N_{01}^{ij})^2}{(N_{11}^{ij} + N_{10}^{ij})(N_{10}^{ij} + N_{00}^{ij})(N_{00}^{ij} + N_{01}^{ij})(N_{01}^{ij} + N_{11}^{ij})} \tag{5}$$

where N_{11} denotes the number of common vehicles appear both in traffic event E_i and E_j, N_{01} means the number of vehicles only appear in the traffic event E_i, while N_{10} means the number of vehicles only appear in the traffic event E_j, and N_{00} represents the number of vehicles neither appear in traffic event E_i nor traffic event E_j.

We finally obtain the influence probability $P(E_i, E_j)$ between traffic event E_i and E_j as follow in Equation 6

$$P(E_i, E_j) = \alpha SC(E_i, E_j) + \beta TC(E_i, E_j) + \gamma LC(E_i, E_j) \tag{6}$$

where $\alpha + \beta + \gamma = 1$.

C. Independent Cascade Model

In considering operational models for the spread of a traffic event through a city (network), represented by a directed graph G, we will speak of each node as being either *influenced* or *uninfluenced*. Thus, the process will look as follows from the perspective of an initially uninfected node E_i: once the propagation is initiated, more and more E_i's neighbors become infected, and E_i's situation may, in turn, trigger further congestions by nodes to which E_i is connected. Many models have been proposed to capture the diffusion process in our traffic events network. The conceptually simplest model of this type is called *Independent Cascade Model*, which is investigated [10], [16].

The Independent Cascade model can well represent the scenario of our traffic network, and it is very efficient to compute. The process starts with an initial set of active nodes A_0. When node u first becomes active in step t, it is given a chance to activate one of its currently inactive neighbor v with probability $p_{u,v}$ (independently of the history so far). If u succeeds, then v will become active in step $t+1$. The process terminates when no more activations are available.

D. Problem Formalization

We can now formally express the optimization problem.

Definition 2 (J-KEY-Region): Given a set S of traffic events, we define the influence of S, denoted by $\sigma(S)$, as the expected total weight of nodes influenced during the diffusion process initiated by S. Given a traffic event network, $G(V, E)$. We denote V to be the set containing all the nodes in G, each of which corresponds to a traffic event and E to be the set containing all the directed edges in G. Each edge $e_j \in E$ in form of (u, v) is associated with a weight $p_{(u,v)} \in [0, 1]$.

TABLE II
$\sigma(T^*)$ OF ALL POSSIBLE SIZE-2 COMBINATIONS

Regions	Corresponding Events	$\sigma(T^*)$
$\{r_i, r_j\}$	$\{E_1.r_i, E_2.r_i, E_3.r_i, E_1.r_j, E_2.r_j, E_3.r_j\}$	6.2
$\{r_i, r_l\}$	$\{E_1.r_i, E_2.r_i, E_3.r_i, E_1.r_l, E_2.r_l\}$	**7.81**
$\{r_i, r_k\}$	$\{E_1.r_i, E_2.r_i, E_3.r_i, E_1.r_k, E_2.r_k\}$	6.82
$\{r_j, r_l\}$	$\{E_1.r_j, E_2.r_j, E_3.r_j, E_1.r_l, E_2.r_l\}$	6
$\{r_j, r_k\}$	$\{E_1.r_j, E_2.r_j, E_3.r_j, E_1.r_k, E_2.r_k\}$	5.5
$\{r_l, r_k\}$	$\{E_1.r_l, E_2.r_l, E_1.r_k, E_2.r_k\}$	5.3

Given G, an integer J and a region-event relation function

$$\mathbb{F} = \{f, T, R | \forall t \in T, \exists r \in R, f(t) = r\} \tag{7}$$

where T and R represents the set of traffic events and set of regions respectively. *J-KEY-Region problem* asks, for a parameter J, to find a subset $R^* \subseteq R$ of size J, such that $\sigma(T^*)$ is maximized, where $T^* = \{t^* | f(t^*) = r^*, r^* \in R^*\}$.

J-KEY Running Example: We have a set of regions with traffic congestion (candidates) $\{r_i, r_j, r_k, r_l\}$. Each region contains a group of events and the weight between two nodes represents the influence probability between two events. We further assume we want to select the two most influential regions from them. Hence we have $J = 2$, $R = \{r_i, r_j, r_k, r_l\}$. There are C_4^2 possible combinations, each of which indicates a $\sigma(T^*)$, see Table II. The candidate regions with traffic congestions $\{r_i, r_l\}$ is the optimal choice. Because when $\{r_i, r_l\}$ occur congestions, the expectation of the number of regions be influenced to have congestions is maximized. Unfortunately, the J-KEY-Region problem is NP-hard for the *Independent Cascade (IC) model* (*Proof*[1]).

III. APPROXIMATION ALGORITHM

A. Greedy Approach

Our first main result is that the optimal solution for *J-KEY Region* problem can be efficiently approximated to be within a factor of $1 - 1/e - \varepsilon$ in *Independent Cascade* model; here e is the base of the natural logarithm and ε is any positive real number. We discuss our ration approximation on the submodular property of function $\sigma(.)$.

Theorem 1 (Submodularity): Let U be a universe set of regions and S be a subset of U. Function $f(.)$ which maps S to a non-negative value is said to be submodular if given any $S \subseteq U$, it holds for any regions $x_1, x_2 \in U - S$ that $f(S \cup \{x_1\}) + f(S \cup \{x_2\}) \geq f(S \cup \{x_1, x_2\}) + f(S)$ (*Proof*[2]).

Property 1: Function $\sigma(.)$ in J-Key-Region is submodular.

Algorithm 1 achieves the performance guarantee, which is a natural greedy hill-climbing strategy related to the approach

---

[1] https://github.com/solrac1986/optimal_seeds_traffic_congestions/blob/master/models.md

[2] https://github.com/solrac1986/optimal_seeds_traffic_congestions/blob/master/approximation_algorithm_theorem_1.md

```
Require:
    $G(V,E)$: a traffic event network.
    $J$: the maximun size of seed region set.
    $\mathbb{F}$: a region-event relation function,
    $\mathbb{F} = \{f, T, R | \forall t \in T, \exists r \in R, f(t) = r\}$.
    $R$: a region set.
Ensure:
    $R^*$: a seed region set.
$R^* \leftarrow \emptyset$
while $|R^*| < J$ do
    $u \leftarrow argmax_{x \in R-R^*}(\sigma(T^* \cup T_x) - \sigma(T^*)), T^* = \{t^* | f(t^*) \in R^*\}, T_x = \{t | f(t) \in x\}$
    $R^* \leftarrow R^* \cup \{x\}$
end while
return $R^*$
```

Algorithm 1: Greedy Algorithm for J-Key-Region.

considered in [17]. Thus, the main content of this result is the framework needed for obtaining a provable performance guarantee, and the fact that hill-climbing is always within a factor of at least $(1 - 1/e)$ of the optimal in this problem.

B. Adaptive Approach

The adaptive approach is a strategy that chooses the optimal region based on the realization of the previous decision. In general, after selecting each region r (a set of events), we can realize which other events have been influenced by observation. As a result, we are required to adaptively pick a sequence of regions with uncertain outcomes under partial observability, which leads to a stochastic optimization problem. In this Section, we study how to select regions sequentially in such a dynamic scenario.

1) A Dynamic Scenario: First, we will consider the problem where we sequentially pick a region $r \in R$, then observe its influence (i.e., determine the events influenced), then pick the next region r', observe its influence, and so on. After each selection, our observations so far can still be represented as a *Traffic Event Network*, but with some edges having probabilities updated to 0 or 1.

Second, we pursue the stochastic optimization under this dynamic scenario. In the previous section, we found that the function of traffic influence $\sigma(.)$ is submodular. Hence the greedy algorithm achieves a near-optimal solution. This stochastic optimization fulfills the property of *adaptive submodulariy*, generalizing submodular set functions to adaptive policies. As a result, a simple adaptive greedy algorithm in the dynamic scenario is guaranteed to be competitive with the optimal policy.

2) Formulation: Now we formulate the stochastic optimization. Each region $r \in R$ is in a particular state $\Phi(r) \in O$ from a set O of possible states. In other words, O refers to the possible world of the traffic event network G; Φ indicates one of the possible outcomes of the traffic event network; $\Phi(r)$ reveals the state of the incoming edges of r after r is selected. Hereby, $\Phi : R \rightarrow O$ is a random realization of R, indicating in which state each region is. The probability distribution $\mathbb{P}(\Phi)$ over the realizations can be easily computed based on the probabilities associated with the edges.

After selecting each region $r \in R$, our observation so far can be represented as a *partial realization* $\Psi \subseteq R \times O$, a function from some subset of E (i.e., the set of regions that we already selected) to their states. A partial realization is consistent with a realization if they are equal everywhere in the domain of Ψ. In this case, we write $\Phi \sim \Psi$. If Ψ and Ψ' are both consistent with some Φ and $dom(\Psi) \subset dom(\Psi')$, we say Ψ is a sub-realization of Ψ'.

Then an adaptive strategy for selecting regions as a *policy* π, which is a function from a set of partial realizations to R, specifying which region to select next under a particular set of observations. If $\Psi \notin dom(\pi)$ the policy terminates (stops selecting regions) upon observation of Ψ. Then, $E(\pi, \Phi)$ is defined as the set of regions selected by π conditioned on realization Φ. Given a policy π, its expected number of events to be influenced, becomes $f(\pi) := \mathbb{E}_\Phi \sigma(E(\pi, \Phi), \Phi)$. Finally, the goal of the adaptive stochastic maximization problem is formulated as follows:

$$\begin{aligned} \pi^* &\in \arg_\pi \max f(\pi) \\ s.t. &\forall \Phi, |E(\pi, \Phi)| \leq J \end{aligned} \tag{8}$$

3) Adaptive Submodularity: Submodularity, as mentioned before, is an intuitive notion of diminishing returns, which states that adding an element to a small set helps more than adding that same element to a superset. The *adaptive submodularity* extends the property of submodularity to cases where the plan can be changed as new information is incorporated. Recent advances in stochastic optimization have extended the property of submodularity to cases where the plan can be changed as new information is incorporated. Hence, the adaptive monotonicity and submodularity properties are defined in terms of the conditional expected marginal benefit of an item.

In the rest of this Section, we show that function $f(\pi)$ from the optimization problem (formula 8) is adaptive submodular. We demonstrate this result with the following theorem.

Theorem 2 (adaptive Submodularity): The function $f(\pi)$ defined in the stochastic maximization problem defined in formula 8 is adaptive submodular (*Proof*[3]).

As a result, a simple extension of the J-KEY-Region algorithm would achieve a near-optimal solution. We demonstrate the adaptive greedy algorithm - at each iteration, we find the optimal region according to the current traffic event network G. After each region is selected, G is updated according to the new information learned from the seed, i.e., the probabilities of edges are updated 0 or 1.

IV. PRUNING

In this Section, we illustrate our pruning strategy to increase the efficiency of our approach. The influence from node u to node v is defined as $P(u \rightarrow v)$. If there is a path from u to v, the influence diffused from u to v via this path is no more

---

[3] https://github.com/solrac1986/optimal_seeds_traffic_congestions/blob/master/approximation_algorithm_theorem_2.md

than the minimal probability of an edge in this path. That is to say, $P(u \to v) \leq 1 - \prod_{i=1}^{n}[1 - P_i(u \to v)]$, where n is the number of paths from u to v. Then we have:

- u's influence on the whole graph $P(u) \leq \Sigma_v[1 - \prod i = 1^n[1 - P_i(u \to v)]]$.
- Region r's influence on the whole graph $P(R) \leq \Sigma_u \Sigma_v[1 - \prod i = 1^n[1 - P_i(u \to v)]]$.

After sorting regions according to $P(R)$, we can check the most potential region first to find the seed. Furthermore, the probabilistic traffic congestion influence network graph G_R is simplified. Nodes in identical region are merged as a new node. Then, the key issue is to deal with edges. To make it clear, we take nodes from two regions as an example: Region R_1 contains nodes $n_1, n_2, ..., n_{k_1}$, Region R_2 contains nodes $m_1, m_2, ..., m_{k_2}$. For each node pair n_i and m_j, we have diffusion probability $P(m_j|n_i)$. Likewise, we have $P(n_j|m_i)$, for each node node pair m_i and n_j. Consider node m_i, the probability that it is influenced by nodes from $R_1 = 1 - \prod_j[1 - P(m_i|n_j)]$. We add the edge from new node R_1 to R_2 with probability $max_i[1 - \prod_j[1 - P(m_i|n_j)]]$. Similarly we have $R_2 \to R_1$. Every time we pick a possible region as a new seed, we first *estimate* its marginal gain by subtracting the *exact* influence in graph G with *seeds set* S from that of $S \cup R$ in G_R. This *estimated* marginal gain will act as the upper bound of R's *actual* marginal gain. A list of regions is initiated by sorting with *key* $P(R)$. We update the *key* after each time of *estimation*, with *estimated* value of the picked-out region. Every updating is followed by a reordering. Then, with descending order, we *compute* the exact marginal gain of regions in the list. Once a region is *computed*, the *key* is updated again. In the subsequent process, if a *computed* region is picked out, it will be add to the *seeds set* $R*$. Details are shown in Algorithm 2.

V. EXPERIMENT

In this Section, we evaluate both the efficiency and effectiveness of our methods using real-world trajectories obtained from GPS-equipped taxis in Shanghai.

A. Experimental setup

The **dataset** consists of GPS trajectories (frequency of minutes) of 10,200 taxis of Shanghai in two months (May and August) in 2012. The total distance of the dataset is over 300 million kilometers, and the total number of GPS points is almost 460 million. The average sampling interval of the dataset is 80.2 seconds. Using the major roads (there is a road level associated with each edge) from the network, Shanghai has been partitioned into 367 regions.

Hardware. All experiments are conducted on a server equipped with Intel(R) Core(TM)i7 3.40GHz PC and 16GB memory, running on Microsoft Windows 7.

B. Evaluation on Effectiveness

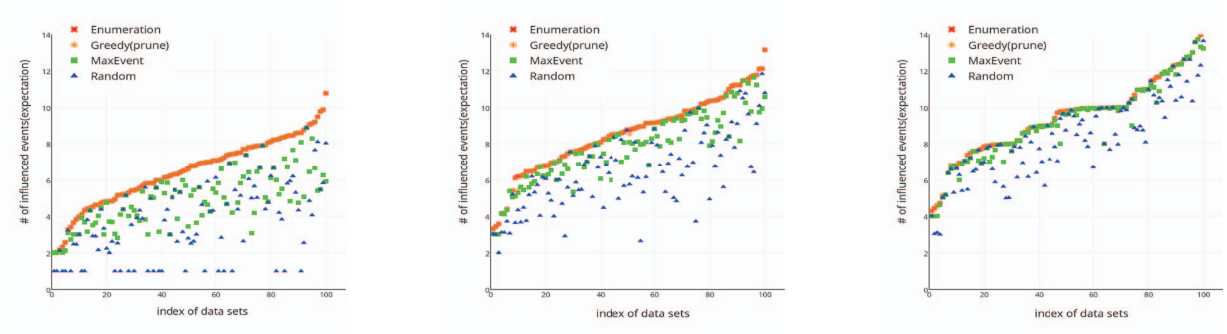
In this subsection, we will evaluate the effectiveness of *greedy* approach and *Adaptive* strategy.

```
Require:
    $G(V, E)$: a traffic event network.
    $J$: the maximun size of seed region set.
    $F$: a region-event relation function,
    $F = \{f, T, R | \forall t \in T, \exists r \in R, f(t) = r\}$.
    $R$: a region set.
Ensure:
    $R^*$: a seed region set.
$R^* \leftarrow \emptyset$
Build priority queue $Q$ for regions in $R$ according to
    approximated $P(R)$.
Build $G_R$.
while $|R^*| < J$ do
    while $Q$ is not empty do
        $r = Q.pop()$
        if $r$ is fresh then
            Estimate($r$) using $G_R$
            add $r$ into $Q$ with new estimated value
        else if $r$ is estimated then
            Compute($r$) using original graph
            add $r$ into $Q$ with new computed value
        else if $r$ is computed then
            add $r$ into $R^*$ and break
        end if
    end while
    mark all elements in $Q$ as fresh
end while
return $R^*$
```

Algorithm 2: Greedy algorithm with pruning.

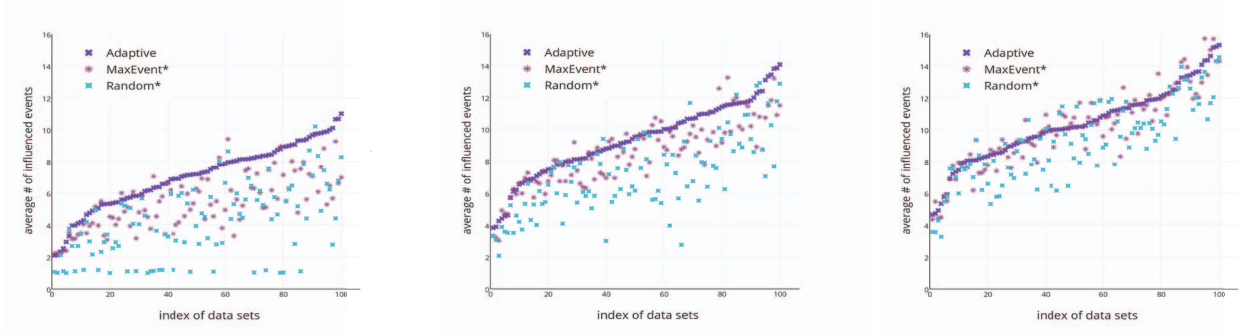
1) Effectiveness of Greedy Approach: We conduct an effectiveness evaluation on our *Greedy* approach and compare our proposed *Greedy* algorithm against three alternative methods: (1) *Enumeration* (optimal), a brute-force algorithm, which computes all C_r^j possible combination of regions and picks the one with maximal expectation of influenced events, where r is the number of candidates regions, and j is the number of key regions to select; (2) *MaxEvent*, an algorithm that chooses the set of regions of size j that contains maximal number of events and pick the regions with the most severe traffic problems; (3) *Random*, the regions are chosen randomly.

Due to the extremely high computational cost of the *Enumeration* algorithm, we only pick ten regions from the original graph as candidate regions and set $|J|$ as 1, 2, 3, respectively. We test 100 sets of candidate regions and report their performance in Figure 1. From the experimental results, we can see that the performance of *Random* can be arbitrarily bad, which is more evident when $|J|$ is relatively small. The *Greedy* algorithm proposed in this paper well approximates the performance of the *Enumeration* (optimal). Because the dataset is small, in our tests, the outputs of *Greedy* are exactly the same as *Enumeration*. This is consistent with our theoretical analysis that *Greedy* performs an approximation of 63%, as shown in the previous section. In addition, *Greedy* outperforms *Random* in most data sets. Also, we can learn that *MaxEvent* outperforms *Random* in some cases, but it is not as good as *Enumeration* / *Greedy*. This further proves our understanding that the region with the most number of traffic congestions is not necessary to be the one which has the most significant influence on the whole traffic condition. In addition,



(a) $|J| = 1$ $|R| = 10$ (b) $|J| = 2$ $|R| = 10$ (c) $|J| = 3$ $|R| = 10$

Fig. 1. Effectiveness of Methods with Various $|J|$ ($|J|$ is the number of regions to be selected).



(a) $|J| = 1$ $|R| = 10$ (b) $|J| = 2$ $|R| = 10$ (c) $|J| = 3$ $|R| = 10$

Fig. 2. Effectiveness of the adaptive approach with various $|J|$ ($|J|$ is the number of regions to be selected). The x-axis denotes the index of 100 data sets, and the y-axis denotes the expectation of several events influenced, which is the objective function.

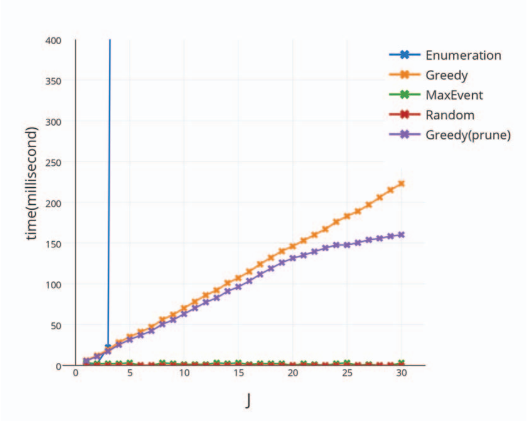


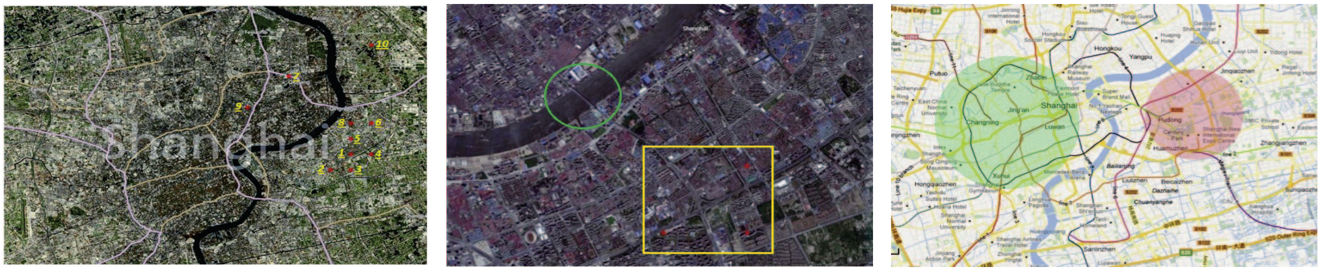
Fig. 3. Efficiency of methods.

we apply our pruning strategy to *Greedy* algorithm, namely *Greedy(prune)*. We find that the performance of *Greedy(prune)* is exactly the same as *Greedy*, which demonstrates that the pruning strategy does not decrease the performance on the effectiveness of the original approach.

2) Effectiveness of Adaptive Approach: We evaluate the effectiveness of *Adaptive* approach. As *Adaptive* approach is a strategy that makes the decision based on the partial realization of the last step, we compare our strategy with the other two strategies:(1) *MaxEvent**: a strategy that chooses region with maximal events after each realization. (2) *Random**: the regions are chosen randomly after each realization. Also, we pick 100 data sets of size ten report their performance in Figure 2 (sorted by the outputs of adaptive approaches). The x-axis denotes the index of 100 data sets, and the y-axis denotes the realization of events that are influenced, which the strategy tries to maximize. We can easily observe that *adaptive* strategy significantly outperforms the *Random** baseline. The performance of *Random** becomes better with the growing of $|J|$, but still worse than our approach. Similarly, *MaxEvent** is generally better than *Random** in most test cases, but cannot equally compete with the performance of our approach.

C. Evaluation on Efficiency

We compare our *Greedy* algorithm with the *Enumeration* algorithm, *MaxEvent* and *Random* algorithms. As shown in Figure 3, the *Enumeration* algorithm (denoted by Enumeration) entails exponential computation time, and the *Greedy* algorithm is much more effective than *Enumeration*. Please note that we stop *Enumeration* after running it over 500 seconds. Although the time cost of *Random* and *MaxEvent* are lower than *Greedy*, the difference is acceptable (within 200ms)



(a) Top 10 seeds. (b) Top 3 seeds. (c) Shanghai subway network.

Fig. 4. Case study of the seeds discovery in Shanghai.

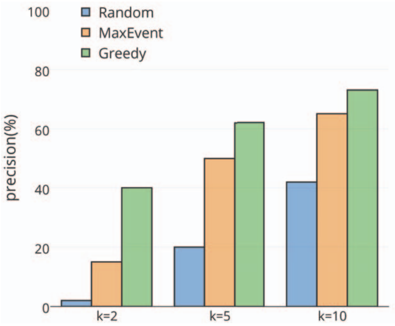


Fig. 5. Precision of J-KEY-Region selection.

even when $|J|$ is relatively large. For the pruning strategy, we can observe that the advantage is more evident with the increasing of $|J|$.

D. A real case study

We conduct a real-data study to verify that our approaches are beneficial for unbar constriction and management. We set the forecast prediction as one hour [18] and determine traffic congestion by judging whether the instantaneous speed of a region is less than 30% of the daily average speed of that region. Based on these settings, we carry out our experiment and get the top 10 seeds as plotted in the first subfigure of Figure 4.

We mark the top-10 seeds in Figure 4a; eight of them lie on the east side of Huangpu River, given the observation that the east side (financial district) covers one-third of the whole city, and the city center is located on the west side. Though in this section, the real situation of the traffic network in Shanghai will be analyzed, and the validity of this result will be proved. Let us take the first three seeds as examples to illustrate the rational analysis.

In Figure 4b we depict centroids of the top-3 seeds (red points) and the Yangpu bridge (circle). East and west parts of the city contain an approximately equal number of taxi cabs, and most of them travel across the river several times a day. The situation is often worse off during rush hours when roads that lead cars to bridges or tunnels are busy and prone to congestion. Therefore, regions where entries to bridges and tunnels have a significant influence on the city's traffic.

Figure 4c presents the roads and subways layout in Shanghai city, from which we can see subways more are concentrated on the west than the east side. Therefore, congestion happens on the east side has a more crucial influence on the transportation system as a whole. These all give the reason why the majority of the top-10 seeds lie on the east side.

Since it was not feasible to get the real ground truth, we crowdsourced[4] a questionnaire where we first conduct a qualification test to understand the respondent's degree knowledge of Shanghai traffic conditions. A score range of $[0, 1]$ is assigned to each respondent to indicate his/her reliability. Then 100 respondents were asked to score the top 30 candidate regions we give in the range $[0, 10]$ to depict its influence on Shanghai's traffic condition when there is traffic congestion. We ranked the 30 candidates regions according to the scores and weighted by their reliability. We compare the top 2, top 5, and top 10 seeds between the questionnaire and our approach results, respectively. The top 10 have relatively high hits, which can reach as high as 73% (Figure 5). For all the scenarios, the precision of our approach outperforms other methods by 12% on average.

VI. RELATED WORK

The increasing volume of trajectory data has led to numerous studies on mobility pattern mining [19], [20], routes [21], average speed [22], and other road-network performance metrics. In [6], [23], the equivalent grid, regions bounded by a major road, and road segment are proposed as candidates. Both the *region-based* and the *road-based* modeling are useful to some extent. The road-based model is more accurate but has a higher cost for computation. Splitting the traffic flow from one road to another probably cannot solve the congestion because the traffic is still in that conservative region. Therefore region should be a more proper unit in urban planning from a macro perspective.

However, regions bounded by roads leave roads lying on boundaries out of account [6]. Seeing that this topic remains open so far, we propose a novel approach to do the second partition based on the result from [6]. After this partitioning, every point on the map is affiliated to the nearest *seed*.

---

[4] http://gmission.github.io/

The rate that major roads lie within boundaries increases. Moreover, the value of *road-based* models is amplified after we discover seeds if they are pipelined subsequently because of the reduction of computation and sparseness. There is no edge in [5], they introduce *Likelihood Ratio Test* directly on nodes matrix. In the literature, we find [24], which quantifies the dependency. However, it is naively modeled as numbers of *Links* between regions, which ignores the three components spatial, temporal, and logical in our work. It has been argued that traffic congestion is not only a condition characterized by slower speeds but also a propagating process. It comes from the observation that the abnormal traffic at one place affects those elsewhere progressively [6].

All state-of-the-art frameworks focus on major cause inferring for a short time span [25], [26], aiming at most anomalous route link [8] or traffic pattern [6]. The *influence* [27] is defined deterministically as frequency of sub-trees from *STOutlier* trees. In the literature, *Independent Cascade* model is used in modeling the diffusion of product adoption in social network [28] and viral market [29].

VII. CONCLUSION

In this paper, we propose a mobility pattern mining system to discover the most influential regions in urban traffic congestions, namely, *seed*. We formalize the *seed discovery* as a problem to find the top J most influential regions, *J-KEY-Region*, and prove it to be NP-hard. Solving this problem has many challenges, which are addressed in this paper by 1) an *event-based* model, which measures the correlation among traffic anomalies from a spatial, temporal and logical perspective; 2) two approximation approaches which employ *Independent Cascade* model. A pruning strategy is applied to improve the effectiveness of our algorithms. Our experimentation verifies the effectiveness and efficiency of our methods with a large scale GPS dataset. Therefore, our approach presents a good potential to be utilized in future smart cities, where vehicles will provide their positions in real-time.

REFERENCES

[1] S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy, "A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 125–151, 2015.

[2] K. Ng, C. Lee, S. Zhang, K. Wu, and W. Ho, "A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion," *Computers & Industrial Engineering*, vol. 109, pp. 151–168, 2017.

[3] M. Mahmoudi and X. Zhou, "Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations," *Transportation Research Part B: Methodological*, vol. 89, pp. 19–42, 2016.

[4] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye, "Mining periodic behaviors for moving objects," in *Proceedings of the 16th ACM SIGKDD*. ACM.

[5] L. X. Pang, S. Chawla, W. Liu, and Y. Zheng, "On detection of emerging anomalous traffic patterns using gps data," *Data & Knowledge Engineering*, vol. 87, pp. 357–373, 2013.

[6] J. Lan, C. Long, R. C. W. Wong, Y. Chen, Y. Fu, D. Guo, S. Liu, Y. Ge, Y. Zhou, and J. Li, "A new framework for traffic anomaly detection," in *2014 SIAM International Conference on Data Mining (SDM'14)*, 2014.

[7] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang, "Hierarchical and networked vehicle surveillance in its: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 25–48, 2017.

[8] S. Chawla, Y. Zheng, and J. Hu, "Inferring the root cause in road traffic anomalies." in *ICDM*, 2012, pp. 141–150.

[9] K. Wong and H. A. Müller, *An efficient implementation of fortune's plane-sweep algorithm for voronoi diagrams*. Citeseer, 1991.

[10] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing letters*, vol. 12, no. 3, pp. 211–223, 2001.

[11] ——, "Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata," *Academy of Marketing Science Review*, vol. 9, no. 3, pp. 1–18, 2001.

[12] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical review E*, vol. 51, no. 2, p. 1035, 1995.

[13] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

[14] A. Rosenfeld, "Connectivity in digital pictures," *Journal of the ACM (JACM)*, vol. 17, no. 1, pp. 146–160, 1970.

[15] T. Pan, A. Sumalee, R.-X. Zhong, and N. Indra-Payoong, "Short-term traffic state prediction based on temporal–spatial correlation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1242–1254, 2013.

[16] H. L. Zhang, J. Liu, C. Feng, C. Pang, T. Li, and J. He, "Complex social network partition for balanced subnetworks," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 4177–4182.

[17] A. Krause and C. Guestrin, "A note on the budgeted maximization of submodular functions," 2005.

[18] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos, "A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 557–569, 2015.

[19] D.-H. Shih, M.-H. Shih, D. C. Yen, and J.-H. Hsu, "Personal mobility pattern mining and anomaly detection in the gps era," *Cartography and Geographic Information Science*, vol. 43, no. 1, pp. 55–67, 2016.

[20] L. Zhang, S. Luo, and H. Xia, "An investigation of intra-urban mobility pattern of taxi passengers," *arXiv preprint arXiv:1612.08378*, 2016.

[21] Y. Ge, H. Xiong, C. Liu, and Z.-H. Zhou, "A taxi driving fraud detection system," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE.

[22] B. Pan, U. Demiryurek, and C. Shahabi, "Utilizing real-world transportation data for accurate traffic prediction." in *ICDM*, 2012, pp. 595–604.

[23] H. Wang, H. Wen, F. Yi, H. Zhu, and L. Sun, "Road traffic anomaly detection via collaborative path inference from gps snippets," *Sensors*, vol. 17, no. 3, p. 550, 2017.

[24] D. Zhang, T. He, F. Zhang, M. Lu, Y. Liu, H. Lee, and S. H. Son, "Carpooling service for large-scale taxicab networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 3, p. 18, 2016.

[25] M. R. Islam, N. I. Shahid, D. T. ul Karim, A. Al Mamun, and M. K. Rhaman, "An efficient algorithm for detecting traffic congestion and a framework for smart traffic control system," in *Advanced Communication Technology (ICACT), 2016 18th International Conference on*. IEEE, 2016, pp. 802–807.

[26] Z. Cao, S. Jiang, J. Zhang, and H. Guo, "A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1958–1973, 2017.

[27] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 807–816.

[28] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD*. ACM.

[29] C. Long and R.-W. Wong, "Minimizing seed set for viral marketing," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE.