Evolutionary Intelligence
https://doi.org/10.1007/s12065-019-00332-4

SPECIAL ISSUE

Mining traffic congestion propagation patterns based on spatio-temporal co-location patterns

Lu Yang[1] · Lizhen Wang[1]

Received: 25 July 2019 / Revised: 27 November 2019 / Accepted: 28 November 2019

Abstract
Traffic congestion is a direct reflection of the imbalance between supply and demand for a certain period of time. Owing to the complexity of traffic roads and the propagation of congestion, the evacuation of traffic congestion for local road sections alone cannot achieve significant results. Based on the measured data of traffic flow, this paper combines the topology of the road network and the existence time of congestion to judge the spatio-temporal correlation of congestion between road sections. We proposed a spatio-temporal co-location congestion pattern mining method to discover the orderly set of roads with congestion propagation in urban traffic, and measure its influence in congestion events. The proposed method not only reveals the process of congestion propagation but also uncovers the main propagation paths leading to the large-scale congestion. Finally, we experimented with the algorithm on the traffic dataset in Guiyang city. The experimental results reveal the traffic congestion rule in Guiyang City, including the prevalent co-occurrence of congestion propagation patterns and their influence in congestion events.

Keywords Spatio-temporal data mining · Traffic congestion propagation pattern · Influence

1 Introduction

With the acceleration of urbanization and the popularity of private cars, the increase in traffic volume makes the urban traffic congestion become serious. It is often seen that there is congestion in the road A, and the adjacent road B also becomes congested after a certain period of time. This congestion is generally a chain reaction caused by peak period, prevalent congestion or some non-cyclical events, etc. The congestion has spatio-temporal relevance and transitivity. The propagation rule of traffic congestion is an important basis for formulating targeted traffic management. However, in traffic congestion events, different treatment schemes for traffic congestion will produce different congestion propagation effects. In the process of congestion propagation, some propagation paths are highly influential, which can make congestion spread more quickly around. For example, A → B is an influential propagation path. Road A is congested and transmits congestion to road B. When both A and B are congested, congestion will spread more quickly. Therefore, identifying the propagation path that plays a key role in congestion propagation and effective management of relevant roads is an effective way to alleviate the urban traffic congestion.

Aiming at the problem of Traffic Congestion Propagation in the urban traffic network, the evolution and propagation rule of traffic congestion is studied by simulation methods [1, 2], such as complex network dynamics [4–8], cellular transmission model [9], SIR model [10]. Some studies describe road networks behaviour using fluid-dynamic models [3–6], and [6] provides more explanations about the behaviour of emergency vehicles in a critical situation of traffic. [7] considers a possible global optimization of car traffic networks by considered delayed differential equations for the description of congested parts on roads. [8] considers possible approaches of congestion on telecommunication networks, whose lines are modeled by conservation laws. In order to reduce the impact of network size on the computational performance of the traffic simulation model, Zhang et al. [9] combine the cell transmission model (CTM) with the macroscopic fundamental diagram (MFD) to simulate the spread of traffic congestion throughout the network.

✉ Lizhen Wang
lzhwang@ynu.edu.cn

[1] Department of Computer Science and Engineering, Yunnan University, Kunming 650091, Yunnan, China

Published online: 10 December 2019

In order to overcome the "distortion" problem caused by the theoretical assumptions and parameter settings of the existing simulation-based congestion analysis method. Based on the traffic flow data, many studies have analyzed the causal relationship between congested roads and described the specific propagation process of congestion based on Bayesian Network and Data Mining [11–16]. Liu et al. [11] model the traffic congestion propagation in a new situation where the vehicle path planning is driven by spatial or temporal preference. Hoang Nguyen et al. [12, 13] construct causality trees from congestion and estimate their propagation probabilities based on Dynamic Bayesian Network, and through the frequent substructure of the causality tree to reveal the causal relationship of congestion propagation and potential bottlenecks in the transportation networks. Shan et al. [14] present a congestion propagation path estimation method based on the greedy algorithm and Bayesian-based method to quickly extract these congestion relationships for visual analytics. However, this needs to assume that the impact of congestion is independent of each other, the isolation of continuous roads cannot reveal complete spatial transitivity.

Saeedmanesh and Geroliminis [15] propose a static clustering method to analyze the spatio-temporal correlation between congested sections by dividing heterogeneous networks into isomorphic associated sub-areas. Rempe et al. [16] present a clustering method to identify the spatio-temporal distribution patterns of congestion and reduce a complex traffic network to the parts that are most frequently congested clusters. These methods can find roads or areas with the same evolution of congestion, but less consider the interaction between roads and the transitibility of congestion. [18] uses the spatio-temporal co-location pattern mining to discover congestion propagation patterns. Spatio-temporal co-location (co-occurrence) pattern mining extends the mining task to the scope of both space and time [19–23]. [19] defined the mixed-drove pattern that a group of mixed object-types often occur in spatial and temporal proximity. This approach carries out the mining process on each time segment, but does not take into account the lifetime of an object. [20] aims to discover patterns that present in sufficient number of time slots with respect to the lifetime of objects. Considering the impact of the time interval between spatio-temporal events, [21] proposed a weighted sliding window model that executes the algorithm in time segments of a given window size. However, the propagation time between congested roads is related to their spatial distance. We define the spatio-temporal neighboring relationship for the delay and directionality of congestion propagation. For different attributes of data, [22] extends their algorithm to handle all feature types (points, lines, polygons) in the mining process of complex applications such as air pollution, [23] identify spatio-temporal co-occurring patterns for continuously evolving spatio-temporal events that have polygon-like representations. We consider the topological relationship between roads and the speed properties of roads during the mining process for congestion propagation.

There are few previous literature that combines the topological distance between roads and the delay and direction of propagation to measure the relationship between congestion propagation. In addition, they did not mine patterns that have influence in congestion events. On their basis, we consider the order and influence of the pattern. In this paper we focus on mining prevalent co-occurrence and influential congestion propagation patterns, which will make a more rapid spread of congestion to the surrounding. In order to solve these problems and successfully mine prevalent co-occurrence and influential propagation patterns, the following problems need to be solved: (1) Traffic congestion is essentially a kind of traffic state. Congestion propagation can be regarded as the interaction of traffic states in different sections. This correlation between roads is constrained by time and space. (2) Owing to the differences in the impact of various roads on the surrounding area or the entire road network, even the impact of each road at different time periods is different, the impact of congestion on different roads and the same roads in different periods of time cannot be regarded as equivalent.

Specifically, this paper introduces spatio-temporal co-locations into the mining of congestion propagation patterns based on the spatio-temporal characteristics of traffic data. More specifically, the contributions we have made in this paper are: (1) Using the topological distance of congested roads on the traffic network and their existence time as constraints to measure the relationship between congestion roads. (2) In order to measure the influence of congestion patterns at various times. The congestion data is divided into congestion event sets according to the spatial–temporal proximity relationship, and according to neighbors affected by the congested road and the neighbor's neighbor to measure the pattern's influence in each event. Then, using the pattern's participation index and propagate influence as the interest measure. In addition, we propose an algorithm for mining influential propagation patterns, and a number of experiments are carried out on the Guiyang traffic data set to verify the effectiveness and practicability of the proposed method.

2 Related definitions

A spatial co-location pattern is a set of spatial features whose instances are frequently located together [17, 24, 25] use distance to measure proximity between instances and the Participation Index (PI) to measure the prevalence of co-location patterns, where the PI is defined as the minimum of the participation ratios (PR) which are the fraction of the number of instances of features forming co-location instances to the total number of instances [26]. On the basis of traditional spatial co-location pattern mining, the time factor is considered to realize spatial–temporal co-location pattern mining [18]. In this section, we present the basic concepts for modeling influential propagation patterns.

Definition 1 (*spatial features*) Spatial features represent different roads in traffic data sets, which is denoted as $F=\{f_1,f_2, \ldots, f_n\}$, where n represents the number of all spatial features.

Taking the road as the spatial feature, In Fig. 1, $F=\{$A, B, C, D, E, F, G$\}$ represents a set of seven different roads.

Definition 2 (*congestion instance*) Taking the dynamic relative deviation rate of road speed as a measurement of traffic state. In Formula (1) $\bar{v}_{f_i t_j}$ is the average speed of the road f_i (the spatial feature f_i) at the time t_j, and v_{fi} is the free-flow speed of the road. Traffic state is congested when ρ is not less than a given congestion threshold *c_ threshold.* A congestion instance on the road represents that it is in a continuous congestion state during a period of time. $f_i{\cdot}j^{(\mathrm{T}[tb,\ te])}$ is the jth congestion instance of the spatial feature f_i, where t_b and t_e are the start and end time of the instance respectively, and T is the date of the instance, the instance represents the traffic state of f_i during the time period $[t_b, t_e]$ of the date T is congestion.

$$\rho = \frac{v_{f_i} - \bar{v}_{f_i t_j}}{v_{f_i}} \tag{1}$$

Definition 3 (*spatial proximity relation SR*) The spatial topological graph takes the spatial feature as the node and the connection relationship between the spatial features as the edge. The spatial distance *sd* of two instances o_i and o_j is the number of edges that the shortest path of their spatial features in the topological graph. When two instances are reachable by e edges on the topological graph and e is not greater than a given spatial proximity threshold *s_threshold*, we say that they satisfy the spatial proximity relation $SR(o_{i,}o_j)$.

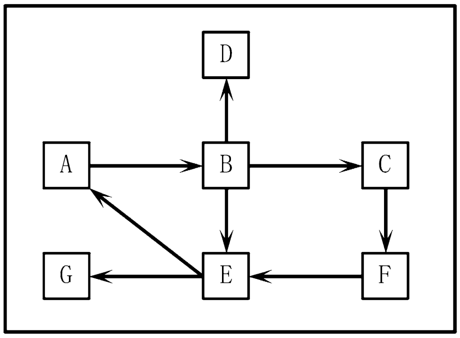

Fig. 1 A spatial topological graph of the road set (the feature set) {A, B, …, G}

Definition 4 (*temporal proximity relation TR*) Given a temporal proximity threshold *t_threshold*. Considering the delay of congestion propagation, given the threshold of time span Δt, if instances $f_{i1}{\cdot}j^{(\mathrm{T}[tb1,\ te1])}$and $f_{i2}{\cdot}k^{(\mathrm{T}[tb2,\ te2])}$ $(i1 \neq i2,\ tb1 < tb2)$ are reachable through e edges on the topological graph, their time distance *td* is calculated by the overlap rate of the duration of congestion. When the overlap rate is not less than the given threshold *t_threshold*, we say that they have a temporal proximity relationship (denoted by $TR(f_{i1}{\cdot}j^{(\mathrm{T}[tb1,\ te1])} \rightarrow f_{i2}{\cdot}k^{(\mathrm{T}[tb2,\ te2])})$), i.e.,

$$\begin{aligned} & td(f_{i1} \cdot j^{(T_1[t_{b1},t_{e1}])}, f_{i2} \cdot k^{(T_1[t_{b2},t_{e2}])}) \\ & \quad = \frac{[t_{b1} + e *\Delta\ t, t_{e1} + e *\Delta\ t] \cap [t_{b2}, t_{e2}]}{[t_{b1} + e *\Delta\ t, t_{e1} + e *\Delta\ t] \cup [t_{b2}, t_{e2}]} \\ & \quad \geq \frac{\min(len([t_{b1} + e *\Delta\ t, t_{e1} + e *\Delta\ t]), len([t_{b2}, t_{e2}]))}{\max(len([t_{b1} + e *\Delta\ t, t_{e1} + e *\Delta\ t]), len([t_{b2}, t_{e2}]))} \\ & \quad * t\_threshold \end{aligned} \tag{2}$$

According to Definitions 3 and 4, when two instances o_i and o_j satisfy both spatial proximity relation $SR(o_{i,}o_j)$ and temporal proximity relation $TR(o_i{\rightarrow}o_j)$, we say that they have spatio-temporal proximity relation $STR(o_i{\rightarrow}o_j)$. It indicates that there is an interaction between traffic congestion states of two instances. That is:

$$STR(o_i \rightarrow o_j) \Leftrightarrow SR(o_i, o_j) \wedge TR(o_i \rightarrow o_j) \tag{3}$$

For example, $\mathrm{A1}^{2019.5.1[7:00,7:10]}$ and $\mathrm{C1}^{2019.5.1[7:16,7:28]}$ are two instances with different features and the same date. The features they belong to can be reached by 2 edges in the given spatial topological graph of Fig. 1. Suppose $\Delta t=7$, $t\_threshold=0.5$ and $s\_threshold=4$. <1> and <2> represent that two instances satisfy both the spatial proximity relation and the temporal proximity relation, so they have spatio-temporal proximity relation $STR(\mathrm{A1}^{2019.5.1[7:00,7:10]} \rightarrow \mathrm{C1}^{2019.5.1[7:16,7:28]})$.

<1> $\quad 2 < s\_threshold = 4;$

<2> $$\frac{[2 * 7, 10 + 2 * 7] \cap [16, 28]}{[2 * 7, 10 + 2 * 7] \cup [16, 28]} \geq \frac{\min(10, 12)}{\max(10, 12)} * 0.5$$

Definition 5 (*Congestion propagation pattern, CPP*) Given a set of spatial features $F=\{f_1, f_2, \ldots, f_n\}$, the congestion propagation pattern is an ordered subset of spatial features whose instances are frequently co-occur together, and it consists of an ordered set of non-repetitive spatial features.

For example, <A, B, C> is a size-3 CPP.

In order to measure the influence of patterns in different congested event sets, we divide the instances according to the neighboring relationships. Directed Neighbor Graph $NG=(V, E)$ is used to represent the relationship between the set of instances, where V is the set of congestion instances, E is the set of edges, and $e_{i,j}=(v_i,v_j)\in E$ represents instances v_i and v_j has a spatio-temporal proximity relation. The partitioned congestion data is represented as connected components in NG, **congestion event set** is recorded as $CES=\{g_1, g_2, \ldots, g_n\}$, where $g_i \in CES$ is the connected component of NG, which is expressed as each congestion event. The directed graph shown in Fig. 2 is generated from neighbors of congestion instances and $CES=\{g_1, g_2, g_3\}$.

Definition 6 (*Row instance and table instance*) In a given set of congestion instances, there exists an ordered set $I=<o_1, o_2, \ldots, o_m>$, where $<o_2, o_3, \ldots, o_{m-1}>$ is a path from o_1 to o_m. If I contains all the spatial features of CPP c and no subset does so, then I is called a row instance (denoted by row_instance(c)) of c. The table instance (denoted by table_instance(c)) of CPP c is the collection of all row instances of c.

For example, in Fig. 2, <A1, B1, C1> is a row instance of $c=$<A, B, C>.

Definition 7 (*Neighbors of instances*) Neighbors of an instance are classified according to the path length between the instance and its neighbors. In $CES=\{g_1, g_2, \ldots, g_n\}$, the shortest path from o_i to o_j can be represented by a set of neighbor relational pairs as $STR(o_i \rightarrow o_{v1})$, $STR(o_{v1} \rightarrow o_{v2})$, …, $STR(o_{v(h-1)} \rightarrow o_j)$, if its length is h, o_j is a h-hop neighbor of o_i. All neighbors of o_i are $IN(o_i) = \bigcap_{x=1}^{m} x - neighbor(o_i)$, where m represents the longest neighbor hops of o_i, x-neighbor$(o_i)=\{o_j| \ STR(o_i \rightarrow o_{v1}), \ STR(o_{v1} \rightarrow o_{v2}), \ \ldots, \ STR(o_{v(x-1)} \rightarrow o_j)\}$.

$l=<o_1, o_2, \ldots, o_k>$ is a row instance of $c=<f_1, f_2, \ldots, f_k>$, whose neighbors are a collection of neighbors of all instances in l, recorded as $LN(l) = \bigcap_{i=1}^{k} IN(o_i)$. If o_j is p-hop neighbor of o_x and q-hop neighbor of o_y, $(o_x, o_y \in l)$, o_j is the min(p, q) hop neighbor of l.

In Fig. 2, one of the shortest paths of A1 to D2 is <A1 → B1 → D2> and the length is 2, then D2 is the 2-hop neighbors of A1. D2 is 1-hop neighbor of B1, so for the row instance <A1, B1> of $c=$<A, B>, D2 is its 1-hop neighbor.

Definition 8 (*The influence of CPP*) Given a set of congestion events $CES=\{g_1, g_2, \ldots, g_n\}$, a size-$k$ CPP c, and its table instance $\mathrm{TI}=\{l_1, l_2, \ldots, l_t\}$. The influence of CPP is determined by the instances it affects and the neighbors of these instances. The steps to measure the influence of CPPs are the following two steps:

(1) The first step is to partition TI based on $CES=\{g_1, g_2, \ldots, g_n\}$, l_a and l_b are classified as the same partition set if l_a, l_b $(l_a, l_b \in \mathrm{TI})$ belong to the same congestion event g_i $(g_i \in CES)$. The partitioned table instances are recorded as $\mathrm{TI}(c)=\{tc_1, tc_2, \ldots, tc_m\}(tc_i \cap tc_j = \emptyset)$.

(2) Calculate the influence of each partitioned set in TI(c), The influence of $tc_i=\{l_1, l_2, \ldots, l_a\}(tc_i \in \mathrm{TI}(c))$ is $TCI(tc_i) = \sum_{i=1}^{k} w_i |i\_neighbor(tc_i)|$, where w_i is the weight of i-hop neighbors, which decreases with the

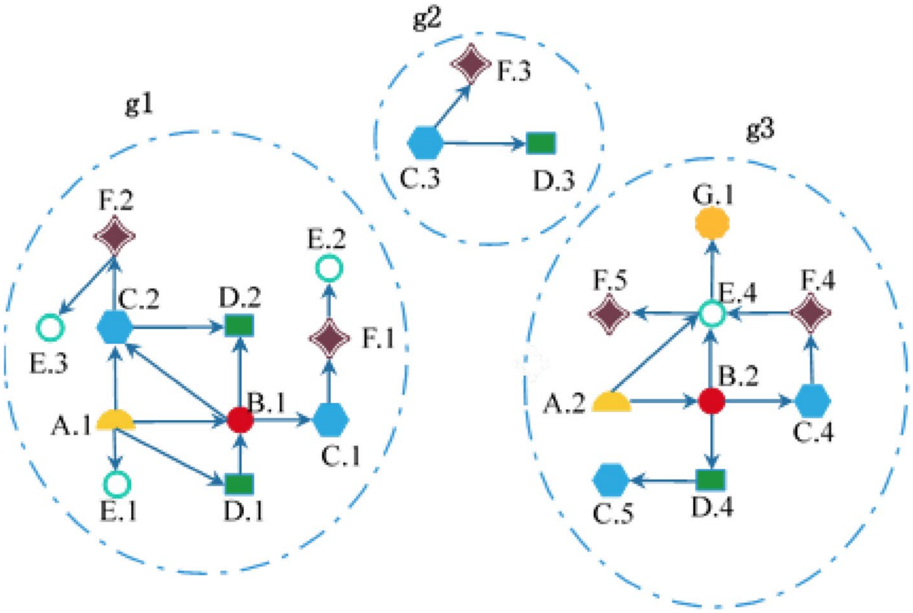

Fig. 2 An example of congestion instances

increase of hops, indicating that the influence on the instance decreases with the increase of hops. The influence of CPP c is $\mathrm{CI}(c) = \min(\mathrm{TCI}(tc_1), \mathrm{TCI}(tc_2), \ldots, \mathrm{TCI}(tc_m))$.

Definition 9 (*Strong congestion propagation pattern, SCPP*) Given a size-k congestion propagation pattern c, a prevalence threshold min_prev and an influence threshold min_inf. c is an SCPP if both of the following conditions are satisfied.

(1) $\mathrm{PI}(c) \geq min\_prev$
(2) $\mathrm{CI}(c) \geq min\_inf$

Take Fig. 2 as an example, $F = \{\mathrm{A, B, C, D, E, F, G}\}$, The number of instances of features A to G are 2, 2, 5, 4, 4, 5, 1, respectively. Set $w_i = 1/i$, $min\_prev = 0.2$, $min\_inf = 4$, $c1 = <\mathrm{A, B, C}>$ has three row instances, (l_1, $l_2 \in g_1$, $l_3 \in g_2$), their neighbors are shown in Table 1. The divided table instance is $\mathrm{TI}(c1) = \{tc_1, tc_2\}$, $\mathrm{TCI}(tc_1) = 5+1 = 6$, $\mathrm{TCI}(tc_2) = 4+3/2 = 5$, $\mathrm{CI}(c1) = \min(5,6) > min\_inf$, and $\mathrm{PI}(c1) = \min(1,1,3/5) > min\_prev$, then $c1 = <\mathrm{A, B, C}>$ is an SCPP. Similarly, in Table 2, $c2 = <\mathrm{C, F, E}>$, $\mathrm{PI}(c2) = 0.6 > min\_prev$, and $\mathrm{CI}(c2) = 3 < min\_inf$, so $c2 = <\mathrm{C, F, E}>$ is not an SCPP.

Lemma 1 (Anti-monotony) *The participation ratio and participation index are anti-monotonic, the influence of CPP and CPP's partition subset are not monotonic.*

Proof For a k-size co-location $c_k = \{f_1, f_2, \ldots, f_k\}$ and $c_{k+1} = \{f_1, f_2, \ldots, f_k, f_{k+1}\}$, assuming that instance o_i is included in the row instance of c_{k+1}, then o_i must also be included in c_k, but o_j in the row instance of c_k is not necessarily in the row instance of c_{k+1}, so the participation ratio is anti-monotonic.

$$\begin{aligned} PI(c_k \cup f_{k+1}) &= \min_{i=1}^{k+1}\{PR(c_k \cup f_{k+1}, f_i)\} \\ &\leq \min_{i=1}^{k}\{PR(c_k \cup f_{k+1}, f_i)\} \\ &\leq \min_{i=1}^{k}\{PR(c_k, f_i)\} = PI(c_k), \end{aligned}$$

so PI is also anti-monotonic. In Fig. 2, CI(B, C) = min(4, 5), CI(C, D) = min(1.5, 1), CI(B, C, D) = 3.3, so CI and TCI are not monotonic.

Table 1 Table instance of pattern $c = <\mathrm{A, B, C}>$ in Fig. 2

Congestion event	Number	Row instance	Neighbors of row instances

g_1	l_1	A.1, B.1, C.1	1_*neighbor*{D.1, D.2, E.1, F.1, F.2}
			2_*neighbor*{E.2, E.3}

	l_2	A.1, B.1, C.2	

g_3	l_3	A.2, B.2, C.4	1_*neighbor*{D.4, E.4, F.4}
			2_*neighbor*{C.5, F.5, G.1}

Table 2 Table instance of pattern $c = <\mathrm{C, E, F}>$ in Fig. 2

Congestion event	Number	Row instance	Neighbors of row instances
g_1	l_1	C.2, F.2, E.3	1_*neighbor*{D.2}
	l_2	C.1, F.1, E.2	
g_3	l_3	C.4, F.4, E.4	1_*neighbor*{G.1, F.4}

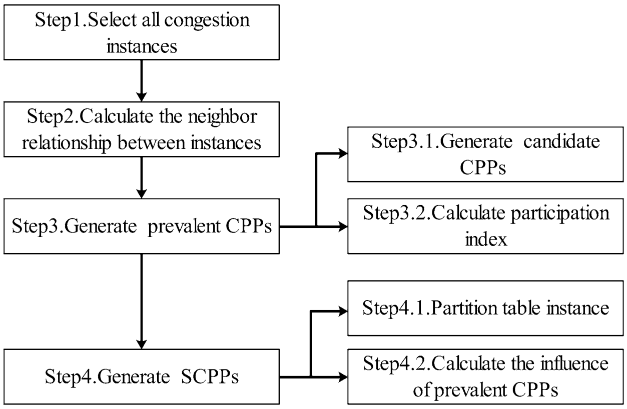
Step1.Select all congestion instances
Step2.Calculate the neighbor relationship between instances
Step3.Generate prevalent CPPs
Step3.1.Generate candidate CPPs
Step3.2.Calculate participation index
Step4.Generate SCPPs
Step4.1.Partition table instance
Step4.2.Calculate the influence of prevalent CPPs

Fig. 3 Method of mining SCPPs

3 Algorithm

This section proposes an algorithm to mine all SCPPs. The main steps of the algorithm are shown in Fig. 3.

First, select the congestion instances from the traffic dataset. Then calculate the spatio-temporal neighboring relationship between instances. The third is to generate CPPs that meet the user's given prevalence threshold. Finally, calculate the influence of prevalent CPPs and select SCPPs that meet the influence threshold.

The proposed algorithm for mining all SCPPs is as follows:

Algorithm

Input: *F*: A set of spatial features; *S*: A set of congestion instances; *TG*: A spatial topological graph; Δ*t*: A time span threshold; *t_threshold*: A temporal proximity threshold; *s_threshold*: A spatial proximity threshold; *min_prev*: A prevalence threshold; *min_inf*: An influence threshold

Output: A set of all SCPPs whose participation index PI(*c*)≥*min_prev* and

influence CI(*c*)≥*min_inf.*

Variables:

$IN = \{IN_1, IN_2, \ldots, IN_n\}$: A neighbor set of instances; $CES = \{g_1, g_2, \ldots, g_n\}$: Congestion event set; *k*: The size of patterns; C_k: Set of candidate size-*k* CPPs; T_C_k: Set of table instances of C_k; P_k: Set of prevalent size-*k* CPPs; T_P_k: Set of table instances of P_k; SP_k: Set of size-*k* SCPPs

Steps：

```
1)  IN = gen_instance_neighborhoods (F, TG, S, t_threshold, s_threshold, Δt);
2)  GES = instances_partition (IN)
3)  k=1;
4)  While ($P_k$≠Ø) do {
5)        If k=1：
6)              $C_2$, $T\_C_2$=gen_2_candidate(IN)
7)        Else：
8)              $C_{k+1}$, $T\_C_{k+1}$=gen_candidate_cpp($P_k$, $T\_P_k$)
9)        $P_{k+1}$, $T\_P_{k+1}$=select_prevalent_cpp($T\_C_k$,min_prev)
10)       $SP_k$=gen_scpp($T\_P_k$, CES, min_inf)
11)       k=k+1}
12) Return ∪ ($SP_2$, …, $SP_k$)
```

Step 1 (Line 1) is to determine the neighborhood relationship between instances of different features according to Δ*t*, *t_threshold*, and *s_threshold* given by the user, so as to find the neighborhood set of each instance. Step 2 (Line 2) divides congestion instances into congestion event sets with adjacent relationships. Step 3 (Line 5–9) is to generate candidate CPPs and determine its prevalence, size 2 candidate CPPs are generated by *IN*; when $k > 2$, size $k+1$ candidate CPPs are generated by size *k* prevalent CPPs. For example, $c1_k$=<f_{a1}, f_{a2}, …f_{ak}> and $c2_k$=<fb_1, fb_2,…,fb_k>, if $fa_2=fb_1$, $fa_3=fb_2$, …$fa_k=fb_{(k-1)}$ and $fa_1 \neq fb_k$ connect $c1_k$ and $c2_k$ to generate size $k+1$ candidate CPP c_{k+1}={*fa*1, *fa*2, …, *fak*, *fbk*}. Step 4 (Line 10) is to calculate the influence of prevalent CPPs according to the partitioned congestion event set, select CPPs with its influence greater than *min_inf* and finally obtains SCPPs.

```
Procedure gen_scpp ($T\_P_k$, $CES$, $IN$, $min\_inf$)

Input:
$T\_P_k$: Set of table instances of $P_k$; $CES=\{g_1, g_2,\dots,g_n\}$: Congestion event set;
$min\_inf$: An influence threshold

Output: A set of SCPPs whose influence CI($c$)≥$min\_inf$.

Variables:
$t\_p_k$: A table instance in $T\_P_k$; $l_i$: A row instance in $t\_p_k$; $partition\_t= \{key_1:tc_1, key_2:tc_2, \dots, key_n:tc_n\}$: Store the partitioned set of $t\_p_k$, where $key$ is the number and $tc_i$ is the partition set $i$;

Steps：

10.1) for each $t\_p_k \in T\_P_k$:
10.2)     for each $l_i \in t\_p_k$:
10.3)         $l_i \in g_j$
10.4)         $key = g_j$,
10.5)         if $key$ in $partition\_t.keys()$:
10.6)             $partition\_t[key]$.append($l_i$)
10.7)         else:
10.8)             $partition\_t[key] = [l_i]$
10.9)     for each $t_{ci} \in partition\_t$:
10.10)        CI($t\_p_k$) = $calculate\_influence(IN)$
10.11)        if CI($t\_p_k$) >$min\_inf$:
10.12)            $SP_k \leftarrow t\_p_k$
10.13) Return $SP_k$
```

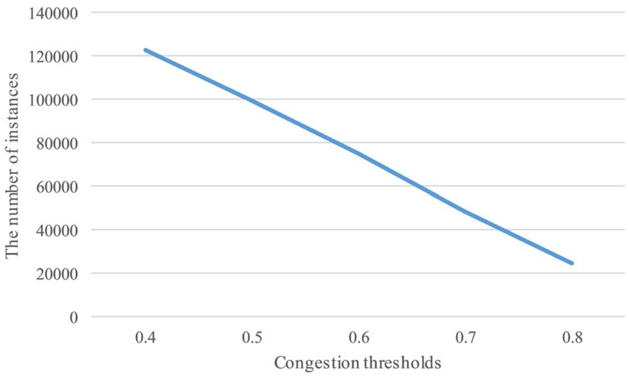

Fig. 4 p impacts on the number of instances

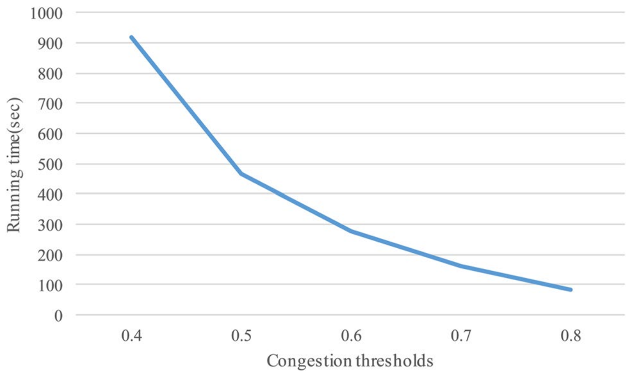

Fig. 5 p impacts on running time

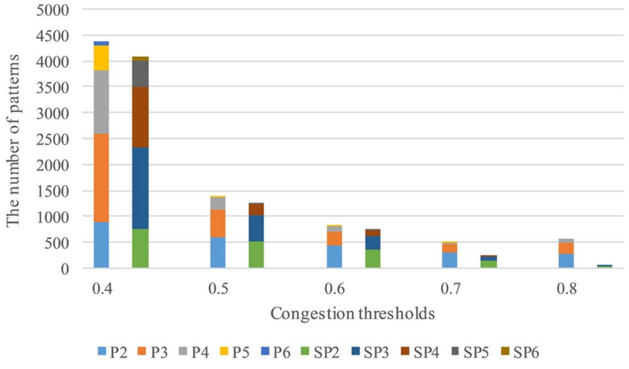

Fig. 6 *p* impacts on mining results

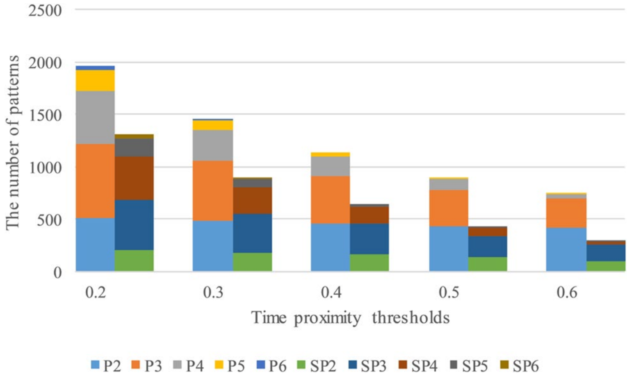

Fig. 8 *t_threshold* impacts on mining results

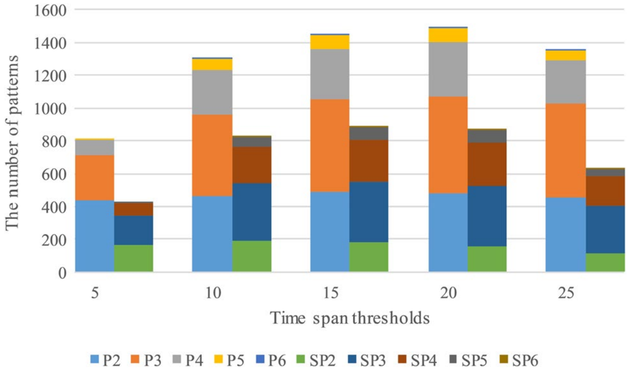

Fig. 7 Δt impacts on mining results

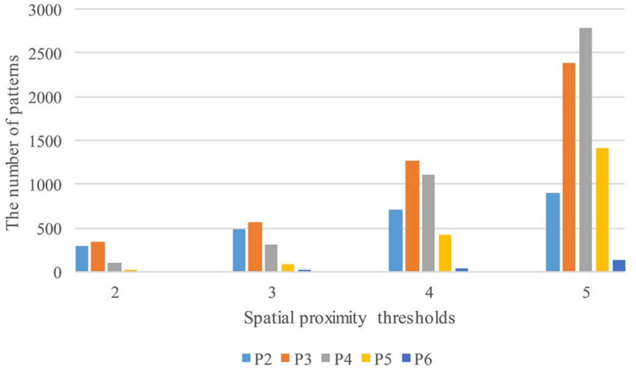

Fig. 9 *s_threshold* impacts on mining results

Procedure gen_scpp is to mine SCPPs from prevalent CPPs. Steps (10.2–10.8) divide the row instances of CPP *c* into one partition set and classifies the row instances of the same event set into one partition set. For example, *c* = <A, B, C > in Table 1, whose row instances are divided into two subsets: $\{l_1, l_2\}$ and $\{l_3\}$. Steps (10.9–10.12) are to calculate the influence of *c*. The influence of CPP is determined by the instances it affects and the neighbors of the instances. The function *calculate_influence* is used to calculate the influence of each subset. The calculation method of CPP's influence is defined in Definition 8. The influence of *c* is the minimum. If the influence of CPP *c* is greater than *min_inf*, then *c* is an SCPP.

Completeness of the algorithm: Step 1 calculates the neighbor relationship by Definitions 3 and 4, and obtaining all neighbor pairs of the instance. Step 6 generates all size 2 candidate patterns according to the neighbor pairs. So there is no loss of size 2 patterns. Step 8 is to generate a size 3 or higher patterns, and the size *k* patterns generated by joining the prevalent size $k-1$ patterns are complete because of Lemma 1. Step 9 guarantees completeness by correctly calculating the participation index of the pattern. Procedure gen_scpp (T_P_k, *CES, IN, min_inf*) is to correctly calculate the influence of these patterns according to Definition 8. So algorithm ensures that all patterns satisfying influence and prevalence are obtained.

Correctness of the algorithm: The correctness of SCPPs can be guaranteed by Steps 9 and (10.10–10.12). Step 9 selects all CPPs that satisfy the user's given prevalence threshold. Steps (10.10–10.12) select all patterns that satisfy the user-defined influence threshold from the CPP obtained in Step 9. So all the patterns returned by the algorithm satisfy the influence and the prevalence.

4 Experimental verification

In this section, we design a series of experiments on real data sets to evaluate the impact of different parameters on the mining results of the algorithm, and evaluate the accuracy of the algorithm. We present the results of CPPs and SCPPs on different parameters and record the number of patterns of different sizes in each mining result, p_k is the set of prevalent size-*k* CPPs and sp_k is the set of size-*k* SCPPs. APSTCP algorithm [18] is also evaluated as a comparison, which is the paper that can be found the closest to our

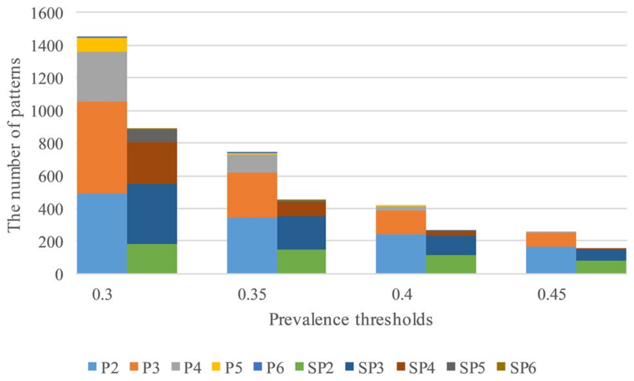

Fig. 10 *min_prev* impacts on mining results

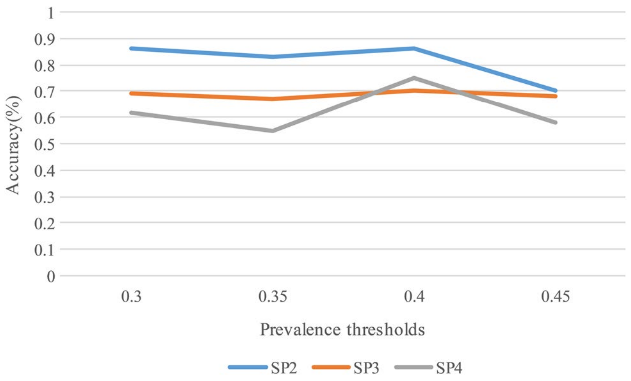

Fig. 12 Accuracy assessment under *min_prev*

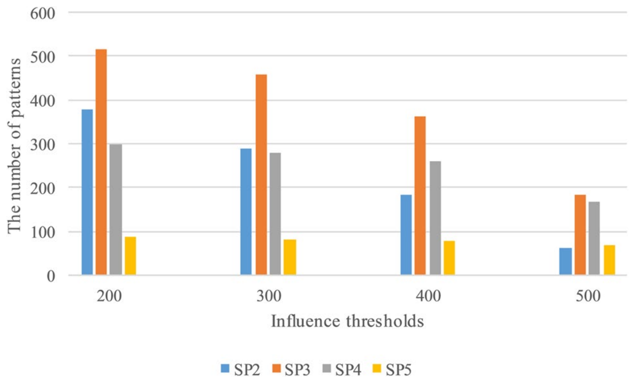

Fig. 11 *min_inf* impacts on mining results

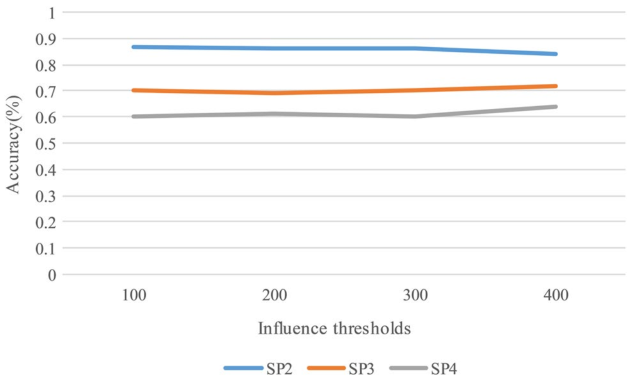

Fig. 13 Accuracy assessment under *min_inf*

work. The APSTCP algorithm considers the spatio-temporal neighbor relationship and uses instance-lookup scheme to identify clique instances. We run all experiments on PC with 4 GB memory and 3.10 GHz Intel(R) Core (TM)i3-2100 CPU. The experimental code is all implemented by python, and the version of Python is 3.6.

4.1 Data

The data comes from Ali Tianchi Intelligent Traffic Forecasting Challenge Competition. Mobile APP is used to collect user's geographic information anonymously in real-time. The data are processed and fused to generate the traffic information of the city at a full time without blind spots. The data contains the attribute information of urban key sections, the network topology structure between sections and the travel time of each section in each historical period. The data used in the experiments are from 01/05/2017 to 30/06/2017, it contains historical travel information for 132 roads with a topological structure as shown in Fig. 18. The data is recorded every 2 min, and the total record is 5,135,845.

4.2 The influence of different parameters on mining results

Experiment 1 analyzes and compares the effects of different congestion thresholds p on the number of congestion instances and the mining results of prevalent CPPs and SCPPs. The other parameters are set to $s\_threshold = 3$, $t\_threshold = 0.3$, $min\_prev = 0.3$, $\Delta t = 3$ and $min\_inf = 100$, The experimental data is from 01/06/2017 to 30/06/2017, the number of instances and the number of patterns are changed when $p = 0.4$, $p = 0.5$, $p = 0.6$ and $p = 0.7$, respectively. It can be seen from Figs. 4, 5 and 6 that as the p decreases, the number of congestion instances increases, and the running time of the algorithm also increases. When p is small, a large number of non-congested instances will be generated, a large number of neighbor pairs will be generated, and the number of indirect neighbors of the instance will also increase, which makes it needs more time to find indirect neighbors when calculating the influence of SCPPs. Therefore, the running time of the algorithm increases as p decreases. In order to timely confine the congestion before large-scale congestion occurs, we choose the congestion threshold of 0.6 to do the next experiment.

Experiment 2 evaluates the effects of different time span thresholds Δ*t* on mining results of prevalent CPPs and SCPPs. The method of controlling variables is adopted in the experiment. In Fig. 7, *t_threshold* = 0.3, *s_threshold* = 3, *min_prev* = 0.3 and *min_inf* = 400. The changes in the number of patterns when Δ*t* = 5, Δ*t* = 10, Δ*t* = 15, Δ*t* = 20, and Δ*t* = 25 are tested respectively. If Δ*t* is too large or too small, it will cause the instance to lose some of its neighbors. As can be seen from the figure, before Δ*t* = 15, the number of patterns increases with the increase of time span, and after that, as the time span threshold increases, the number of patterns begins to decrease. The best effect was obtained at Δ*t* = 15 in the experiment.

Experiment 3 evaluates the effects of different time proximity thresholds on mining results. In Fig. 8, Δ*t* = 15, *s_threshold* = 3, *min_prev* = 0.3 and *min_inf* = 400. The changes in the number of patterns when *t_threshold* = 0.2, *t_threshold* = 0.3, *t_threshold* = 0.4, *t_threshold* = 0.5 and *t_threshold* = 0.6 are tested respectively. The interaction between congestion instances is affected by time and space. When the value of *t_threshold* is larger, the stricter constraints on time and space lead to a decrease in the number of neighboring instances, which reduces the number of patterns.

Experiment 4 evaluates the effects of different spatial proximity thresholds on mining results. In Fig. 9, as the *s_threshold* increases, the number of patterns increases, because when *s_threshold* increases, the number of neighbors increases, and thus the mined patterns increases. The growth rate of the lower size patterns is relatively small, and the growth rate of the higher size patterns is relatively large because when *s_threshold* is increased, spatial features can generate more higher size patterns through the join step.

Experiment 5 evaluates the effects of different prevalence thresholds on mining results. In Fig. 10, *s_threshold* = 3, *t_threshold* = 0.3, *min_inf* = 400 and Δ*t* = 15. It can be seen from the figure that the number of CPPs and SCPPs decreases with the increase of *min_prev*, and the number of prevalent CPPs is much larger than that of SCPPs under the same *min_prev*. With the increase of *min_prev*, the proportion of lower size patterns in CPPs becomes larger and larger, but the number of lower size patterns in SCPPs is the least. This is because the prevalence and influence of SCPPs are considered at the same time. The participation of lower size CPPs is higher, but its influence in congestion events is relatively lower. In order to better reflect the real situation, there are enough patterns to prepare for the subsequent analysis, *min_prev* can't be too big. At the same time, in order to make the pattern as universal as possible, *min_prev* can't be too small, so set *min_prev* as 0.3 in our experiments generally.

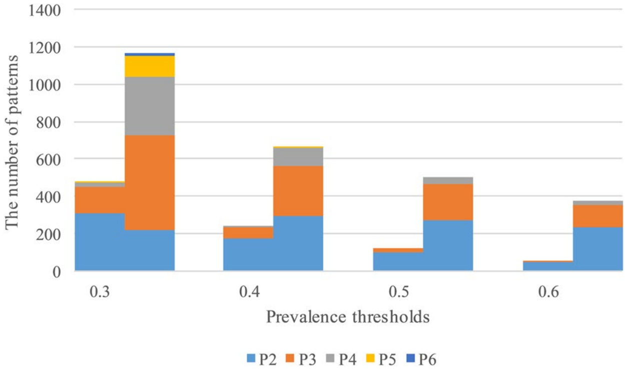

Fig. 15 *min_prev* impacts on mining results

Experiment 6 evaluates the effects of different influence thresholds on mining results. In Fig. 11, *s_threshold* = 3, *t_threshold* = 0.3, *min_prev* = 0.3 and Δ*t* = 15. The changes in the number of patterns when *min_inf* = 200, *min_inf* = 300, *min_inf* = 400 and *min_inf* = 500 are experimented respectively. With the increase of *min_inf*, the number of patterns decreases. As can be seen from the figure, the number of higher size and lower size influential propagation patterns is

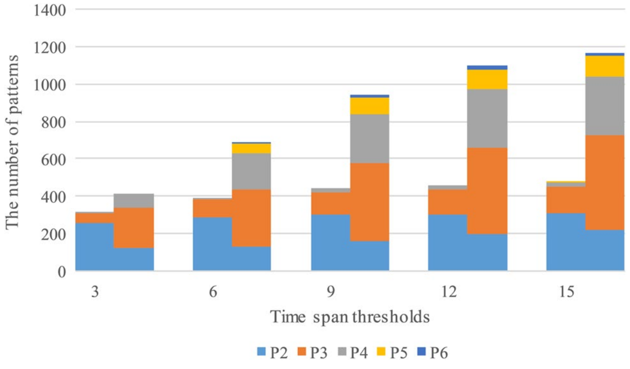

Fig. 14 Δ*t* impacts on mining results

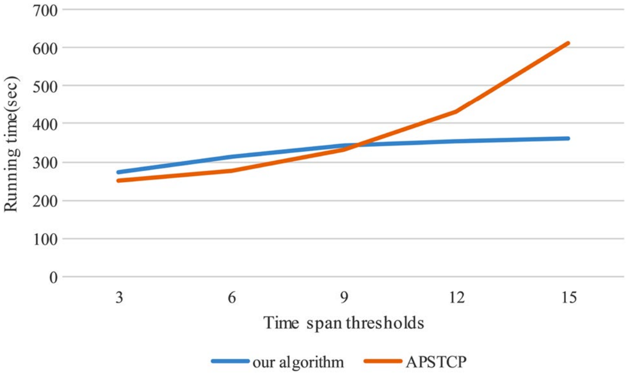

Fig. 16 Δ*t* impacts on running time

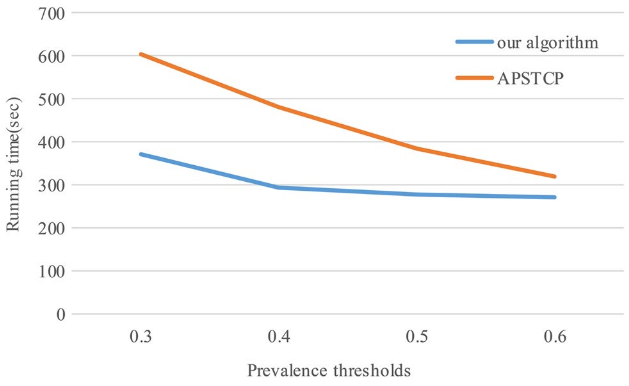

Fig. 17 *min_prev* impacts on running time

relatively small, and the proportion of middle-size influential propagation patterns is the largest. This is because when the scope of congestion is larger, the impact on the surrounding area is more significant, while the influence range of the congestion is within a certain range.

Experiments 7 and 8 mainly evaluate the accuracy of the algorithm. Two parameters *min_prev* and *min_inf* required to screen SCPPs from CPPs were selected to evaluate the accuracy of the algorithm, and the accuracy of the algorithm was verified on Guiyang traffic data. Data includes D1(from 01/05/2017 to 30/05/2017) and D2 (from 01/06/2017 to 30/06/2017). Firstly, mine the SCPP on D1 using the given parameters. Secondly, the number of patterns m that appears again in D2 is calculated. Finally, determine if the reproduced SCPPs are still an SCPP and calculate the number of SCPPs n. The accuracy of the algorithm indicates the probability of being an influential pattern when the pattern reappears, which is equal to n/m. In Fig. 12, $s\_threshold = 3$, $t\_threshold = 0.3$, $min\_inf = 200$ and $\Delta t = 15$. In Fig. 13, $s\_threshold = 3$, $t\_threshold = 0.3$, $min\_prev = 0.3$ and $\Delta t = 15$. As can be seen from the figures, a lower *min_prev* will lead to the lower prevalence of CPPs and lead to

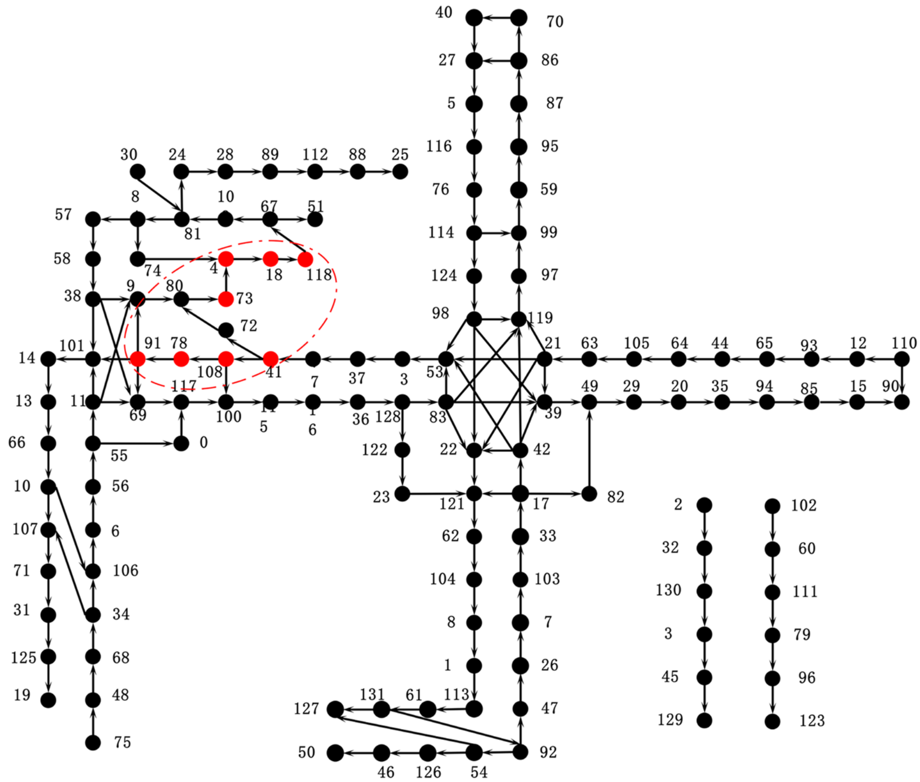

Fig. 18 The road network of Guiyang

Table 3 The top 10 SCPPs in June 1

SCPPs	*x* (days)	*y* (days)	SCPPs	*x* (days)	*y* (days)
<108,41,73,4>	14	13	<108,41,73,91>	14	14
<108,41,73>	28	25	<78,108,41,73>	11	9
<108,41,73,118>	6	6	<41,73,118>	11	11
<108,41,73,18>	16	16	<41,73,4>	22	19
<41,73>	30	25	<41,73,18>	25	23

infrequent SCPPs; a higher *min_prev* will lead to the omission of some prevalent CPPs, resulting in incomplete SCPPs; The accuracy of the higher size patterns is smaller than that of the lower size patterns because the SCPP considers both prevalence and influence, the higher size pattern satisfies the influence threshold but its *min_prev* is lower.

4.3 Comparison with the APSTCP algorithm

Experiment 9 compares our algorithm with the APSTCP algorithm [18]. The experiment measures the traffic state with the same congestion threshold. The experimental data is from 01/06/2017 to 30/06/2017. The number of instances of our algorithm is 74,813, and the number of instances of APSTCP is 337,955. The amount of data in our algorithm is small because an instance time period is the start time and end time of continuous congestion, and each instance of APSTCP is 2 min. The experiment sets the parameters to $s\_threshold = 3$, $t\_threshold = 0.3$ and $p = 0.6$, and compares the same and different mining results of the two algorithms under the same parameters Δt and *min_prev*. In Figs. 14 and 15, the left side of each group in the bar chart is the number of the same results and the right side is the number of different results.

It can be seen from Fig. 14 that as Δt increases, the number of patterns increases, and the patterns with the same mining results are mainly lower size patterns, while the patterns with different results are mainly higher size patterns. This is because APSTCP considers the clique when generating higher size patterns, and it prunes the pattern that does not satisfy the clique. However, our algorithm did not consider the clique but considered the order and transitivity of the congestion instance, thus producing more different higher size patterns.

In Fig. 16, we see that the running time of our algorithm does not fluctuate greatly with the change of Δt. This is because the number of adjacent instances of the instance of our algorithm increases first and then decreases as Δt increases. APSTCP will increase the running time as Δt increases, because when Δt increases, its number of adjacent instances will also increase, resulting in an increase in patterns.

Our algorithm is more sensitive to *min_prev*. When *min_prev* is increased, the number of SCPPs is much reduced. This is because SCPP is ordered. In order to mine the propagation rule of congestion, the order of instance pairs is considered. The spatio-temporal proximity of the two instances is directional, which makes the pattern more but the *min_prev* is smaller. Therefore, when the *min_prev* is smaller, a large number of patterns are easily generated, and when the *min_prev* is larger, many patterns are cut off. It can be seen from Fig. 17 that as *min_prev* increases, the running time of both algorithms decreases, and the gap between them becomes smaller because the number of prevalent patterns generated is less under the constraint of the higher *min_prev*. And most of them are lower size patterns.

In addition, our algorithm also considers the influence of patterns in congestion events, aiming to mine patterns that satisfy both prevalence and influence thresholds. We selected the top 10 patterns of influence rankings in the SCPPs on June 1 for analysis, their topological relationship is shown in the red circle of Fig. 18. The black dot represents road and the number beside the black dot represents no. of the road in Fig. 18. In Table 3, the spatial features are represented by numbers, x represents the number of days in which the *min_prev* of the pattern is greater than 0.3 in June, and y represents the number of days in which the *min_prev* and *min_inf* of the pattern in June are greater than 0.3 and 400, respectively. It can be seen from the table that the frequency of occurrence of the higher pattern is not very high, but when it occurs, the influence of propagation is likely to be greater than the influence threshold. We can also see the propagation law and direction between features from the pattern in the table. The main spatial features of the top 10 patterns of influence on this day are 108, 41, 73, which means that they are the main cause of congestion spreading to the surroundings. Through these patterns, we can provide advice on traffic improvement to reduce congestion.

5 Conclusions

In the actual congestion situation, road congestion is often propagating, a road congestion will cause many roads around it also appear congestion. The prevalent co-occurrence and influential patterns selected from many congestion propagation paths can provide targeted management and grooming schemes for the urban traffic management. So the mining method of influential propagation patterns is proposed. Firstly, according to the spatio-temporal characteristics of the traffic congestion, this paper introduces a spatio-temporal co-location mining method to mine the prevalent co-occurrence of congestion propagation patterns. And the road topology and the existence time of congestion are used as constraints to measure the relationship between the congestion roads. Secondly, considering the difference in the influence of each road on the surrounding roads, the influence of propagation patterns is taken into account when measuring pattern prevalence and influence. Based on the spatio-temporal proximity relation, congestion instances are partitioned and the influence of patterns in different congestion events is measured. Ultimately, the prevalent and influential propagation patterns will be discovered. Our algorithms have been validated by experiments on the Guiyang traffic data set. The experimental results reveal the traffic congestion rules in Guiyang City, including the prevalent co-occurrence of congestion propagation patterns and their influence in congestion events. In the future research, we will supplement the algorithm by considering the regularity and occasionality of congestion.

Acknowledgement This work is supported by the National Natural Science Foundation of China (61966036, 61662086), the Natural Science Foundation of Yunnan Province (2016FA026), and the Project of Innovative Research Team of Yunnan Province (2018HC019).

References

1. Kerner BS (2012) The physics of traffic: empirical freeway pattern features, engineering applications, and theory. Springer, Berlin
2. Daganzo C, Daganzo CF (1997) Fundamentals of transportation and traffic operations. Pergamon, Oxford
3. Garavello M, Piccoli B (2006) Traffic flow on networks. American institute of mathematical sciences, Springfield
4. Cascone A, D'Apice C, Piccoli B, Rarità L (2008) Circulation of car traffic in congested urban areas. Commun Math Sci 6(3):765–784
5. Cutolo A, De Nicola C, Manzo R, Rarità L (2012) Optimal paths on urban networks using travelling times prevision. Model Simul Eng 2012(3):1–9
6. Manzo R, Piccoli B, Rarità L (2012) Optimal distribution of traffic flows in emergency cases. Eur J Appl Math 23(4):515–535
7. Rarità L, D'Apice C, Piccoli B, Helbing D (2010) Sensitivity analysis of permeability parameters for flows on Barcelona networks. J Differ Equ 249(12):3110–3131
8. Cascone A, Marigo A, Piccoli B, Rarità L (2010) Decentralized optimal routing for packets flow on data networks. Discrete Contin Dyn Syst Ser B 13(1):59–78
9. Zhang Z, Wolshon B, Dixit VV (2015) Integration of a cell transmission model and macroscopic fundamental diagram: network aggregation for dynamic traffic models. Transp Res Part C Emerg Technol 2015(55):298–309
10. Zeng Z, Li T (2018) Analyzing congestion propagation on urban rail transit oversaturated conditions: a framework based on SIR Epidemic Model. Urban Rail Transit 4(3):130–140
11. Liu Z, Liu Y, Wang J, Deng W (2016) Modeling and simulating traffic congestion propagation in connected vehicles driven by temporal and spatial preference. Wirel Netw 22(4):1121–1131
12. Liu W, Zheng Y, Chawla S, Yuan J, Xing X (2011) 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, pp 1010–1018
13. Nguyen H, Liu W, Chen F (2016) Discovering congestion propagation patterns in spatio-temporal traffic data. IEEE Trans Big Data 3(2):169–180
14. Shan Z, Pan Z, Li F, Xu H, Li J (2018) Visual analytics of traffic congestion propagation path with large scale camera data. Chin J Electron 27(5):934–941
15. Saeedmanesh M, Geroliminis N (2017) Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. Transp Res Proc 23:962–979
16. Rempe F, Huber G, Bogenberger K (2016) Spatio-temporal congestion patterns in urban traffic networks. Transp Res Proc 15:513–524
17. Wang L, Bao X, Zhou L (2017) Redundancy reduction for prevalent co-location patterns. IEEE Trans Knowl Data Eng 30(1):142–155
18. He Y, Wang L, Fang Y, Li Y (2018) Discovering congestion propagation patterns by co-location pattern mining. In: Asia-Pacific web (APWeb) and web-age information management (WAIM) joint international conference on web and big data. Springer, Cham, pp 46–55
19. Celik M, Shekhar S, Rogers JP, Shine JA (2008) Mixed-drove spatiotemporal co-occurrence pattern mining. IEEE Trans Knowl Data Eng 20(10):1322–1335
20. Celik M (2015) Partial spatio-temporal co-occurrence pattern mining. Knowl Inf Syst 44(1):27–49
21. Qian F, Yin L, He Q, He J (2009) Mining spatio-temporal co-location patterns with weighted sliding window. In: 2009 IEEE international conference on intelligent computing and intelligent systems, vol 3. IEEE, pp 181–185
22. Akbari M, Samadzadegan F, Weibel R (2015) A generic regional spatio-temporal co-occurrence pattern mining model: a case study for air pollution. J Geogr Syst 17(3):249–274
23. Pillai KG, Angryk RA, Banda JM, Schuh MA, Wylie T (2012) Spatio-temporal co-occurrence pattern mining in data sets with evolving regions. In: 2012 IEEE 12th international conference on data mining workshops. IEEE, pp 805–812
24. Wang L, Bao X, Chen H, Cao L (2018) Effective lossless condensed representation and discovery of spatial co-location patterns. Inf Sci 2018(436–437):197–213
25. Bao X, Wang L (2019) A clique-based approach for co-location pattern mining. Inf Sci 2019(490):244–264
26. Wang L, Bao X, Zhou L, Chen H (2019) Mining maximal sub-prevalent co-location patterns. World Wide Web 22(5):1971–1997

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.