Detecting Urban Black Holes Based on Human Mobility Data

Liang Hong [1], Yu Zheng [2*], Duncan Yung [3], Jingbo Shang [4], Lei Zou [5]

[1] *Wuhan University, Wuhan, China,* hong@whu.edu.cn

[2] *Microsoft Research, Beijing, China; Shanghai Jiao Tong University, Shanghai, China,* yuzheng@microsoft.com

[3] *University of Pittsburgh, USA,* duncanyung@cs.pitt.edu

[4] *Shanghai Jiao Tong University, Shanghai, China,* shangjingbo@apex.sjtu.edu.cn

[5] *Peking University, Beijing, China; Key Laboratory of Computational Linguistics (PKU), Ministry of Education, China,* zoulei@pku.edu.cn

ABSTRACT

Many types of human mobility data, such as flows of taxicabs, card swiping data of subways, bike trip data and Call Details Records (CDR), can be modeled by a *Spatio-Temporal Graph* (STG). STG is a directed graph in which vertices and edges are associated with spatio-temporal properties (e.g. the traffic flow on a road and the geospatial location of an intersection). In this paper, we instantly detect interesting phenomena, entitled black holes and volcanos, from an STG. Specifically, a black hole is a subgraph (of an STG) that has the overall inflow greater than the overall outflow by a threshold, while a volcano is a subgraph with the overall outflow greater than the overall inflow by a threshold (detecting volcanos from an STG is proved to be equivalent to the detection of black holes). The online detection of black holes/volcanos can timely reflect anomalous events, such as disasters, catastrophic accidents, and therefore help keep public safety. The patterns of black holes/volcanos and the relations between them reveal human mobility patterns in a city, thus help formulate a better city planning or improve a system's operation efficiency. Based on a well-designed STG index, we propose a two-step black hole detection algorithm: The first step identifies a set of candidate grid cells to start from; the second step expands an initial edge in a candidate cell to a black hole and prunes other candidate cells after a black hole is detected. Then, we adapt this detection algorithm to a continuous black hole detection scenario. We evaluate our method based on Beijing taxicab data and the bike trip data in New York, finding urban anomalies and human mobility patterns.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-data mining, Spatial Databases and GIS

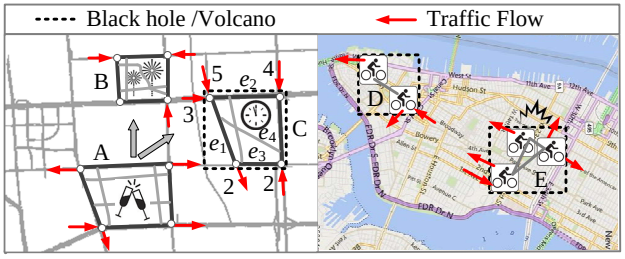
Keywords

Urban computing, Spatio-temporal graph, Black hole detection

1. INTRODUCTION

Advances in sensing technology have lead to a huge amount of human mobility data [24], such as flows of taxicabs, card swiping data of subways, bike trip data and Call Details Records (CDR), which can be modeled by a *Spatio-Temporal Graph*. A Spatio-Temporal Graph (STG) is a directed graph in which vertices and edges have geospatial positions and spatial lengths respectively, and are associated with spatio-temporal attributes. For example, as shown in Fig. 1(a), in a road network, POIs (Point Of Interests) with geographical positions can be regarded as vertices, and road segments between POIs can be modeled as edges. Traffic flows between POIs always change dynamically over time, and edges can be removed from the graph due to a traffic control or a disaster. Likewise, the bike sharing system illustrated in Fig. 1(b) can be modeled as an STG, where bike stations can be regarded as vertices and the connections between them as edges. The cycling users form the traffic flows between bike stations.

In an STG, we can find interesting phenomena, e.g. black holes and volcanos, which may represent urban anomalies or people's regular travel patterns. Specifically, a black hole is a subgraph of an STG that has the overall inflow greater than the overall outflow by a threshold, while a volcano is such a subgraph that has the overall outflow greater than the overall inflow by a threshold. As different applications usually have different spatial constraint on a black hole or volcano(e.g. what if an entire city is a black hole in a bike sharing system), a spatial threshold on the size of a black hole is usually needed during the detection.



(a) Stampede in Shanghai (b) NYC Bike Sharing System

Figure 1: Two Examples of Black Holes and Volcanos

Example 1: As illustrated in Fig. 1(a), based on taxicab data and CDR data, when the 2015 new year countdown in Shanghai was approaching, the majority of people started to leave the bar street to the Bund, forming a volcano *A*; 1 hour later many of them will enter a firework display viewing region and a countdown plaza, which is comprised of a collection of road segments and neighborhoods, resulting in two black holes *B* and *C* respectively. A catastrophic stampede then happened mainly because an unexpected volume of traffic streaming into these black holes without enough spaces. This can actually be captured by the black hole detection a few hours *before* the tragedy. Specifically, once the delta between the inflow and outflow (defined as *actual inflow* in Definition 3) is approaching the upper bound of road network capacity in the region, the information about black holes can be displayed on street-side

*Yu Zheng is the correspondence author

SIGSPATIAL '15 November 03 - 06 2015, Bellevue, WA, USA

DOI: http://dx.doi.org/10.1145/2578726.2578744.

screens to suggest nearby drivers not going to the Bund. In addition, police can consider a traffic control around the black holes and suggest visitors to leave the places.

Example 2: Based on bike trip data of NYC bike sharing system Citibike, the inflow and outflow of a bike station can be obtained by counting people returning and renting bikes at the station, respectively. When a temporary traffic control, shown in Fig. 1(b), was taken due to a traffic accident, many people within the affected area chose to rent bikes as an alternative means of transportation, forming a volcano *E*. With the real-time information of *E*, Citibike can temporarily use trucks to deliver bikes to the stations in *E* to alleviate the shortage of bikes. As shown in Fig. 1(b), a volcano *D* is regularly formed around Wall street during everyday rush hour, because many commuters rent bikes from nearby stations after work. The area of influence (e.g. usually more than one bike station is included in a volcano) and duration of the volcano continuously change in a dynamic STG, which are usually beyond the common knowledge of locals. With the help of this volcano pattern, Citibike can optimize the design of its bike sharing system by adding more bikes and docks to the bike stations around Wall street.

In this paper, we instantly detect black holes and volcanos from an STG. As the detection of volcanos is proved to be equivalent to that of black holes, we only focus on black hole detection in the rest of the paper. This is a very challenging problem, as in reality a black hole is usually a combination of multiple edges and vertices (i.e. subgraph) in a dynamic graph subject to both spatial and flow constraints. Black hole detection is proved to be equivalent to graph clustering (detailed in Section 2.2), which is an NP-complete problem. However, dynamics of an STG's spatio-temporal properties, e.g. the flow changing over time, make our problem more challenging than existing graph clustering problems [10, 9, 19]. Moreover, since both the area of influence and duration of black holes evolve with time, we should detect them timely and continuously. Unfortunately, methods for mining time-evolving graphs [11, 15] do not work well for our problem either (detailed in Section 6).

To this end, we first propose an STG index, based on which an efficient algorithm is proposed to detect black holes from an STG in a time interval. Then, we adapt the detection algorithm to a continuous black hole detection scenario to reduce the computation cost. The contribution of this paper lies in three aspects:

1) When instantly detecting black holes in a time interval, we propose 1) a candidate selection algorithm that finds candidate grid cells to start from and 2) a spatial expansion algorithm that expands an edge in a candidate grid cell to a black hole. An upper bound of a grid cell's actual flow is defined to help select and prune candidate cells after each black hole is detected.

2) We propose a continuous detection algorithm to further reduce the total cost of black hole detection in multiple time intervals, utilizing both detected results in the previous time interval and historical patterns of black hole over a long period.

3) We evaluate our method using Beijing road network and real GPS trajectories generated by over 33,000 taxis, and bike trips generated by over 6,300 bikes in New York City. Two case studies demonstrate that our method can detect black holes/volcanos representing unusual events and human mobility patterns that can improve the urban planning of Beijing and the operational efficiency of NYC bike sharing system. The performance evaluation proves that our method outperforms baseline methods.

2. PRELIMINARIES

2.1 Definitions

DEFINITION 1. *(STG) A Spatio-Temporal Graph (STG) $G = (V, E)$ is a directed graph, where V and E respectively denote the complete set of vertices and edges in G ($|E| \geq 1$). Each vertex $v \in G.V$ has a geospatial position and each edge $e \in G.E$ has a spatial length $e.l$. Each vertex/edge is associated with attributes varying in time, e.g. the inflow and outflow.*

DEFINITION 2. *(Inflow/Outflow) For each edge $e \in G.E$, the inflow of e in time interval t is $f_{in}(e,t) = \sum_{e' \in (G.E-e)} f(e', e, t)$, where $f(e', e, t)$ is the flow from e' to e in t. Likewise, the* outflow *of e in t is $f_{out}(e,t) = \sum_{e' \in (G.E-e)} f(e, e', t)$. Suppose S is a subgraph of an STG G, $S.E \subseteq G.E$ is the collection of edges in S. The* inflow *of S is $f_{in}(S,t) = \sum_{e \in S.E \wedge e' \in (G.E-S.E)} f(e', e, t)$, and outflow is $f_{out}(S,t) = \sum_{e \in S.E \wedge e' \in (G.E-S.E)} f(e, e', t)$.*

DEFINITION 3. *(Actual Flow) In an STG, the actual flow f_a of an edge e is $e.f_a = f_{in}(e,t) - f_{out}(e,t)$ and that of a subgraph S is $S.f_a = f_{in}(S,t) - f_{out}(S,t)$.*

DEFINITION 4. *(Black Hole) In an STG, a subgraph S is a black hole if and only if: $S.f_a \geq \tau$, and $MBB(S) \leq d$, where τ and d are flow and spatial thresholds, respectively. $MBB(S)$ denotes the spatial* minimum bounding box *(MBB) of S.*

DEFINITION 5. *(Volcano) A subgraph S of an STG is a volcano if and only if: $-S.f_a \geq \tau$, and $MBB(S) \leq d$.*

As shown in Fig. 1 (a), in black hole *C*, $e_1.f_a = 5 - 2 = 3$ and $e_4.f_a = 4 + 2 = 6$. The subgraph S formed by e_1, e_2, e_3, and e_4 has an actual flow $S.f_a = 5+4+3+2-2 = 12$ and a MBB with a diagonal $1.2km$. S is a black hole, if $\tau = 10$ and $d = 1.5km$.

In real applications, a black hole usually needs a spatial constraint d, as a very big black hole is not actually useful to solve a problem. For instance, an entire city can be regarded as a black hole in a bike sharing system without given a spatial constraint. This kind of black hole cannot reflect traffic anomalies or people's travel patterns in a city. We use the MBB to control a black hole's spatial range. Other shapes like a disc can also be employed in our framework as a spatial constraint. In the case study, we set d as the length of a MBB's diagonal, which is equivalent to the size of a MBB. τ and d can be set by specific applications. Although how to choose τ is not a focus of our paper, we give one method to set τ in the following. Given a region constrained by d, τ can be the capacity of the region's road network, i.e., $\tau = \alpha \times \sum_{e \in S.E} e.l \times e.n/L$, where $e.n$ is the number of lanes of e, L is the average length of a vehicle (say 4.5 m), and α is a factor $\in [0, 1]$.

2.2 Problem Statement

Given an STG G, a time interval t, a flow threshold τ and a spatial threshold d, the black hole detection in G is to find out a set of subgraphs of G, denoted as $BH = \{S_1, S_2, ..., S_n\}$, such that:

1) $\forall S \in BH$ satisfies Definition 4;

2) $\forall S \in \{G - BH\}$ does not satisfy Definition 4;

3) for $1 \leq i, j \leq n, i \neq j, S_i \cap S_j = \emptyset$, i.e. not connected, and $S_i \cup S_j$ does not satisfy Definition 4.

The reason why we do not generate overlapped black holes is derived from real applications (It does not mean we cannot do that). Presenting many overlapped black holes with minor differences to transportation authorities or end users is not very helpful.

THEOREM 1. *Detection of black holes in an STG is equivalent to that of volcanos.*

PROOF. Suppose $G' = (V', E')$ is an inverse graph of $G = (V, E)$, where $V' = V$ and $\forall e' \in G'.E'$, e' is the corresponding edge of each edge e in $G.E$, $f_{in}(e', t) = f_{out}(e, t)$ and $f_{out}(e', t) = f_{in}(e, t)$. If S is a black hole in G, then $f_{in}(S,t) - f_{out}(S,t) \geq \tau$. Consequently, the corresponding S' in G' has $f_{out}(S', t) - f_{in}(S', t) \geq \tau$, which is the definition of a volcano. Thus, detecting black holes in an STG is equivalent to detecting volcanos in its inverse graph. Similarly, we can prove that detecting volcanos in an STG is equivalent to detecting black holes in its inverse graph. □

THEOREM 2. *Detecting black holes from an STG is NP-Complete.*

PROOF. An STG in a time interval can be regarded as a weighted and directed graph, where the flow on each edge represents the edge weight. Considering the following *local density* graph clustering problem which is known to be NP-complete [17]: given a weighted and directed graph $G = (V, E)$, find out a set of subgraphs from G in which each subgraph S has k vertices (i.e. $|S| = k$) and its local density $\delta_{\text{int}}(S) = \frac{1}{|S|(|S|-1)} \sum_{e \in S.E} e.f_a \geq r$.

Black hole detection problem can be restricted to local density problem by allowing only instances in which d is set to the maximum possible MBB formed by k vertices, and $\tau = r \times k\,(k-1)$. Therefore, local density problem is a special case of black hole detection problem, which proves the NP-completeness of detecting black holes from an STG. □

Due to Theorem 1 and Theorem 2, we propose an approximate solution for detecting black holes. Additionally, even given the same STG and thresholds, black holes with slightly different presentations would be detected in the same geographic region, if using different detection strategies. Though our method has the ability to detect different presentations of black holes by slightly adjusting the expansion strategy, we just provide one of the results in a region in our case study. In real applications, providing many similar black holes in a region is not that useful.

2.3 Framework

Fig. 2 presents the framework of our method, which is comprised of two major components: black hole detection in a single time interval and continuous detection from consecutive time intervals. To deliver a more tangible and concrete story, in the rest of this paper, we use GPS trajectories of vehicles and a road network to perform a case study. Specifically, an edge of an STG and its inflow/outflow are instantiated by a road segment and the flow of vehichles traversing the road segment, respectively. Our method is general to being applied to other data sources, such as bike trip data in a city and Call Details Records, as long as they can form an STG.

Detection in a single time interval: We map the GPS trajectories received in the recent time interval onto a road network using a map-matching algorithm [21], and then calculate the inflow/outflow of each road segment in the interval. Using a spatio-temporal indexing algorithm (detailed in Section 3.1), we partition a city into disjoint grid cells, each of which may cover a few road segments, and build an STG index that maintains the upper bound of the actual flow of each cell and the actual flow of each road segment belonging to the cell. Based on the STG index, the candidate cell selection algorithm finds the candidate cells that could contain a black hole (detailed in Section 3.2). The spatial expansion algorithm starts from the candidate cell that has the maximum actual flow, expanding the road segment with maximum actual inflow in the cell to a black hole. It then adds neighboring segments gradually until the spatial and flow constraints are no longer fulfilled (detailed in Section 3.3). Once a black hole is detected, we recalculate the upper bound of the actual flow for the remaining candidate cells. According to the updated upper bound, more cells can be pruned from the candidate set. The spatial expansion algorithm repeats until all the candidate cells have been checked.

Continuous black hole detection: This component further improves the efficiency and effectiveness of black hole detection by using the knowledge from previous and historical time intervals. This component is motivated by two observations. First, a black hole evolves with time but may not change tremendously in geographic spaces in two consecutive intervals. Second, the occurrence of black holes follows a certain pattern. Thus, in an offline process, we detect the frequent subgraph patterns from the historical black holes that have been detected in the same time interval of different days over a long period. The black holes/volcanos detected from time interval t and the frequent black hole patterns of $t+1$ can become initial graphs for the spatial expansion algorithm to start from in $t+1$. Based on these black holes and volcanos, we can quickly identify new black holes in $t+1$ without re-building a black hole fundamentally (detailed in Section 4).

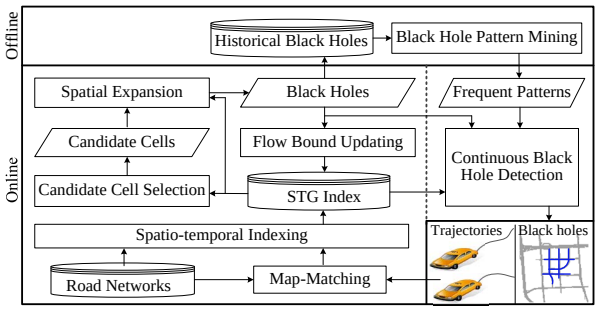


Figure 2: Framework of Black Hole Detection

3. DETECTION IN A TIME INTERVAL

3.1 STG Index

The STG index maintains the structure of an STG, dynamic flows on each edge, and spatial relationships between different edges. Specifically, as shown in Fig. 3, we partition a city into disjoint and uniform grid cells. The spatial relationships between different grid cells are maintained by a matrix. For instance, given the matrix, we can quickly identify the neighborhood of g_{22} is $g_{11} \sim g_{33}$, as illustrated in the left part of Fig. 3. For each grid cell g, we build a list L storing IDs of road segments (i.e. edges) belonging to the grid cell. Each road segment e in $g.L$ is associated with its own actual flow $e.f_a$ of the recent time interval. For each grid cell g, we calculate *positive actual flow* $g.F$ by Equation 1, which only uses the road segments in $g.L$ whose actual flow is *positive*:

$$g.F = \sum\nolimits_{e_i \in g.L \wedge e_i.f_a > 0} e_i.f_a \tag{1}$$

The positive actual flow will be used to calculate the flow upper bound of the cell in the candidate selection step.

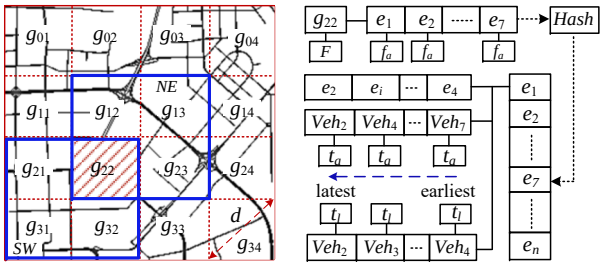


Figure 3: Grid-partitioned STG and STG Index

In the meantime, we build an adjacency list managing the structure and dynamic flow of an STG. For each road segement e in the adjacency list, we maintain three lists. The first is a list of road segments that directly connect to e in the road network; The second is a list of vehicle IDs sorted by their arrival times t_a at e; the third is a list of vehicle IDs sorted by their leaving time t_l from e. The first list is static, while the latter two sorted lists will be updated after each round of map-matching. In a real application, we only need to store the vehicle IDs of recent time intervals, e.g. 1-2 hours. To calculate the actual flow of an edge in a time interval, e.g., $e_7.f_a$, we just need to count the number of vehicles whose arrival time

is within the time interval while the leaving time is outside the interval. The road segment ID in $g.L$ connects to its corresponding records in the adjacency list via a hash function.

Since a road segment may cross two or more grid cells, the STG index can avoid redundant storages and index updates than directly storing everything in a grid cell. To facilitate the later pruning process, the diagonal of a grid cell can be set the same as the spatial constraint of a black hole (see details in the following section).

3.2 Candidate Cell Selection

The candidate cell selection algorithm quickly selects the grid cells that could have a black hole by checking the positive actual flow $g.F$ of each grid cell and that of its neighbors.

Specifically, as shown in Fig. 3, by setting the size of each cell the same as the spatial constraint d of the black hole, we ensure a black hole can simultaneously intersect four grid cells at most. If positive actual flows of the four grid cells are still less than the given flow threshold τ, it is impossible to find any black hole in the four cells (refer to Theorem 3). Given a grid cell, we can check the positive actual flow of the *four-cell combination*s which involves four directions: northwest (*NW*), northeast (*NE*), southeast (*SE*), and southwest (*SW*), as illustrated in the left part of Fig. 3. For each direction, we can obtain a positive actual flow, denoted as $F_{NW}(g)$, $F_{NE}(g)$, $F_{SE}(g)$, and $F_{SW}(g)$, respectively. For example,

$F_{NE}(g_{22}) = g_{12}.F + g_{13}.F + g_{22}.F + g_{23}.F$;

$F_{SW}(g_{22}) = g_{21}.F + g_{22}.F + g_{31}.F + g_{32}.F$;

We define a flow upper bound $UB(g)$ of a grid cell as:

$$UB(g) = Max(F_{NW}(g), F_{NE}(g), F_{SE}(g), F_{SW}(g)) \quad (2)$$

If $UB(g) < \tau$, then g is not a candidate cell. That is, we should not start finding a black hole from g. Based on the STG index, we can quickly calculate $UB(g)$ for each grid cell g, therefore pruning many impossible grid cells. The remaining cells will be used as candidate cells. Though we do not start from the cells that have been pruned in this step, the road segments in these cells would be involved in the following spatial expansion step.

THEOREM 3. *The actual flow of a subgraph $S = (V, E)$ is equal to the sum of the actual flows of its own edges, i.e., $S.f_a = \sum_{e \in S.E} e.f_a$.*

PROOF. According to Definition 3:

$$\sum_{e \in S.E} e.f_a = \sum_{e \in S.E} (f_{in}(e) - f_{out}(e)) = \sum_{e \in S.E} f_{in}(e) - \sum_{e \in S.E} f_{out}(e) = f_{in}(S) - f_{out}(S) = f_a(S) \quad (3)$$

□

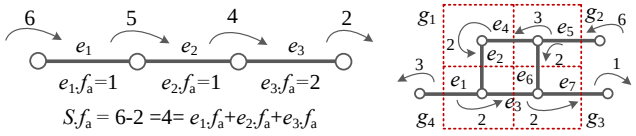


Figure 4: Example of Theorem 3

For example, as shown in the left part of Fig. 4, the actual flow of S is equal to the sum of the actual flows of e_1, e_2, and e_3. Likewise, as illustrated in the right part of Fig. 4, the subgraph S falling in the four grid cells (g_1,g_2,g_3, and g_4) has actual flow $S.f_a = \sum_{j=1}^{7} e_j.f_a = 2$. This is smaller than $\sum_{i=1}^{4} g_i.F = (2+1) + (1+1+1) + (0+2+1) + (2+0) = 11$, because an edge may belong to two grid cells (i.e., the actual flow of the edge will be counted twice) and we do not count negative actual flow when calculating $g.F$. Thus, we prove that for a grid cell g, $F_{NW}(g) \geq \sum_{e \in NW(g)} e.f_a$, $F_{NE}(g) \geq \sum_{e \in NE(g)} e.f_a$, $F_{SW}(g) \geq \sum_{e \in SW(g)} e.f_a$, and $F_{SE}(g) \geq \sum_{e \in SE(g)} e.f_a$. If $\tau > UB(g)$, then $\tau > \sum_{e \in NW(g)} e.f_a$. Thus, the S formed by $e \in NW(g)$ is not a black hole.

3.3 Spatial Expansion

Starting from the candidate cell with the biggest $g.F$, the spatial expansion algorithm expands an initial edge in the cell to a black hole according to the following strategy. The algorithm selects the edge with the largest $e.f_a$ as an initial edge, adding e's neighboring edges one by one to form a black hole. To determine the adding order, we calculate a priority score for each neighboring edge by Equation 4:

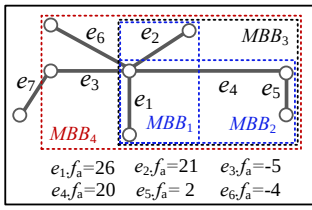
$$R(S,e) = \begin{cases} f_a(e)/\Delta & \Delta \neq 0 \text{ and } e.f_a \geq 0 \\ f_a(e) \times \Delta & \Delta \neq 0 \text{ and } e.f_a < 0 \\ +\infty & \Delta = 0 \text{ and } e.f_a \geq 0 \\ -\infty & \Delta = 0 \text{ and } e.f_a < 0 \end{cases} \quad (4)$$

where e is an edge that we are going to add into an existing black hole S. Δ is defined by Equation 5, denoting the increase of $MBB(S)$ if e is added to S. If S only has one edge, the diagonal of S is set to zero.

$$\Delta = MBB(S+e).diagonal - MBB(S).diagonal \quad (5)$$

The edge with highest priority score is added into the black hole. This score gives a higher priority to the edge that has high actual flow and results in a small increase of MBB area. The edge that violates flow or spatial thresholds cannot be added. The spatial expansion stops until no edge can be added to the black hole.

Fig. 5 presents an example to illustrate the spatial expansion algorithm, supposing the spatial threshold is 7 and the flow threshold is 40. We start from e_1 as it has the highest actual flow, i.e. $S = \{e_1\}$. In reality, such kind of edges usually has a higher probability to form a black hole. Then, we have four neighboring edges (e_2, e_3, e_4, and e_6) to add. Based on Equations 4 and 5, we calculate priority scores for the four edges, respectively. Actually, we do not need to check the MBB of adding e_3 and e_6 at this stage, as these two edges have negative actual flows which lead to low priority scores. Now, we need to compare e_2 with e_4. If adding e_2 into S, the minimal bounding box of S will become MBB_1. Thus, the spatial increase of adding e_2 is $\Delta = MBB_1.diagonal - 0 = 3$, and $R(S, e_2) = \frac{21}{3} = 7$. In contrast, if we add e_4 to S, MBB_2 will become the minimal bounding box of S. Consequently, the score for e_4: $R(S, e_4) = \frac{20}{4} = 5$. As $R(S, e_4) < R(S, e_2)$, e_2 is added into S first. After adding e_2, $S = \{e_1, e_2\}$, $S.f_a = e_1.f_a + e_2.f_a = 47$ and $MBB(S).diagonal = 3$.



	S	MBB(S)	Δ	R(S,e)
1	Ø	0		
2	{e1}	0		
3	{e1,e2}	MBB1=3	3	7
4	{e1,e2,e4}	MBB2=4	2	10
5	{e1,e2,e4,e5}	MBB3=5	0	+∞
6	{e1,e2,e4,e5,e6}	MBB4=7	2	-8

Figure 5: Illustration of the Spatial Expansion Algorithm

Now, e_4 can be added into S, as it has a much higher score than e_3 and e_6; the Δ of e_4 is 5-3=2, and $R(S, e_4) = \frac{20}{2} = 10$. At this moment, $S = \{e_1, e_2, e_4\}$ and $MBB(S) = MBB_3$; Later, e_5 becomes a neighboring edge of S. Because e_5 has a positive actual flow and does not enlarge the spatial range of S, it gets the

highest priority to be added into S (denoted as $+\infty$ in Equation 4). Though e_3 and e_6 have negative actual flows, they would help us as a bridge to find nearby edges with high actual flows. e_6 and e_3 result in the same expansion to $MBB(S)$, while e_6 has a higher score than e_3. Adding e_6 does not break the spatial and flow constraints. However, to control the quality of a black hole, we need to include such kind of edges as fewer as possible. Therefore, after e_6 is added, e_3 will not be added into S (priority score is set to $-\infty$). e_7 cannot be included in the black hole as it breaks the spatial constraint 7. Finally, $S = \{e_1, e_2, e_4, e_5, e_6\}$ is a black hole as $S.f_a = 26 + 21 + 20 + 2 - 4 = 65 > 40$ and $MBB(S).diagonal = 7$.

3.4 Flow Upper Bound Updating

Once a black hole is detected based on the candidate cell with the highest $g.F$, some edges from other candidate cells may have been included in the detected black hole. So, we update the upper bound of the remaining candidate cells by subtracting the actual flows of the edges that have been used in the black hole from $g.F$. Then, we recalculate UB for these candidate cells respectively according to Equation 2, pruning candidate cells whose $UB < \tau$. Later, we choose the candidate cell with maximum $g.F$ from the remaining cells, and perform the spatial expansion algorithm. Once the set of candidate grid cells becomes empty, we stop detecting black holes.

4. CONTINUOUS DETECTION

Since black holes change over time, we need to detect them continuously to inform transportation authority or end users' timely decision-making. To improve the efficiency and effectiveness of our method, we propose a continuous detection algorithm that detects the black holes in consecutive time intervals, utilizing the black holes and volcanoes detected in the past time interval and the black hole patterns mined from a long period of time.

The continuous detection algorithm is motivated by two insights: First, a black hole evolves with time but would not change tremendously in geographic spaces under normal circumstances. It is also interesting to find that a black hole in $t+1$ may originate from a volcano at t, and vice versa. For instance, people watching a football game can form a black hole around a stadium before the game begins, and then form a volcano after the game is over. We can reduce the detection overhead by utilizing black holes detected in time interval t that are still black holes or volcanos in $t+1$. The black holes that changes a lot in two consecutive time intervals, usually in a small number, will be detected from scratch.

Second, the occurrence of black holes follows a certain pattern. For example, a business district is usually a black hole in the morning of a workday.Motivated by the second insight, we use the *gSpan* algorithm [20] to mine the *closed frequent subgraph*s from the historical black holes detected in the same time interval of different days. A graph S is closed in a graph database if there exists no supergraph of S that has the same support as S. Other frequent subgraph mining algorithms can also be applied in our method.

The continuous detection algorithm is comprised of five steps, which have been formally described in Algorithm 1:

1) We detect a set of candidate cells C using the candidate selection algorithm introduced in Section 3.2 (see Line 1).

2) We retrieve the black holes and volcanos detected in time interval t that do not overlap any black hole pattern in $t+1$, and then put them together with the black hole patterns in a union set $BP_{t+1} \cup BV_t$ (see Line 2-5).

To expedite the retrieval process, we organize black hole patterns of each time interval using an R-Tree, as shown in Fig. 6. Given a query black hole/volcano S_q, we first search the R-tree for leaf nodes whose MBBs intersect the minimum bounding box MBB_q of S_q. Then, we quickly check whether these black hole patterns contain the same edge as S_q by doing an XOR operation between the bitmap representations of a black hole pattern and that of S_q. For example, $V_q = 1010000000$ means S_q contains e_1 and e_3.

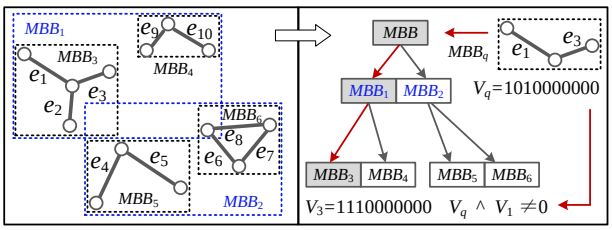


Figure 6: Match a black hole with black hole patterns

3) We check each black hole/volcano S with a positive actual flow in the union set. If $S.f_a \geq \tau$, S is regarded as a new black hole in $t+1$. If $0 < S.f_a < \tau$, we try to find a black hole by adding some neighboring edges with positive actual flows to S, or by removing some non-bridge edges with negative actual flows from S (see Line 6-11). A bridge edge is an edge of a graph whose deletion increases its number of connected components. The method for finding bridge edges in a graph $G = (V, E)$ is widely available and very efficient, which has time complexity $O(|V|+|E|)$, where $|V|$ and $|E|$ are the numbers of vertices and edges in STG G. Additionally, when adding an edge into S, we start from the neighboring edge with a higher priority score than others. On the contrary, the edges with a lower priority score will be removed from S earlier. This step saves the computational load of our method as we do not need to build a black hole fundamentally.

4) Once finding a black hole, we update the flow upper bound of each candidate cell, therefore pruning some cells from C.

5) If C is still not empty, we call spatial expansion algorithm to find more black holes based on C.

Algorithm 1: ContDetection(G, BH_t, BP_{t+1}, d, τ)

Input: BV_t: Black holes and volcanos in t, BP_{t+1}: Black hole patterns of $t+1$, an STG, G, a spatial threshold d and a flow threshold τ.
Output: BH_{t+1}: Black holes detected in $t+1$.

1 $C \leftarrow$ *CandidateSelection* (G, d, τ); $BH_{t+1} \leftarrow \emptyset$;
2 **For** each $S \in BV_t$
3 **If** $\exists S' \in BP_{t+1}$, s.t. $S \cap S' \neq \emptyset$ //non-overlap
4 $BV_t \leftarrow BV_t - S$; //remove a black hole/volcano
5 **For** each $S \in BP_{t+1} \cup BV_t$
6 **If** $S.f_a \geq \tau$ in $t+1$
7 $BH_{t+1} \leftarrow BH_{t+1} + S$;
8 **Else if** $0 < S.f_a < \tau$ in t
9 **While** $MBB(S) < d$
10 **If** S has a neighboring edge with a positive actual flow
11 Add such edges to S according to Equation 4;
12 **Else if** S has non-bridge edges with negative actual flow
13 Remove such edges from S according to Equation 4;
14 **If** $S.f_a \geq \tau$
15 $BH_{t+1} \leftarrow BH_{t+1} + S$;
16 $C \leftarrow$ *UpperBoundUpdating*(S, G); **Continue**;
17 $BH_{t+1} \leftarrow$ *SpatialExpansion*(C, G);
18 **Return** BH_{t+1};

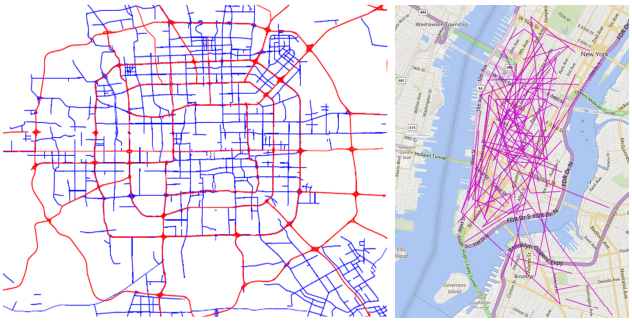
5. EXPERIMENTAL EVALUATION

We first perform two extensive case studies to evaluate the effectiveness of our method. Then, we evaluate the performance of detection in a time interval and continuous detection of our method.

5.1 Data

1) Beijing Taxicab Data: We use the road network data of Beijing, which contains 148,110 road nodes and 96,307 road segments. The total length of all the road segments is 21,895 km, and the spatial area covered by the road network is 2,507km^2. The GPS trajectories were generated by over 33,000 taxis during a period of 30 days in Nov. 2012. The total distance of taxi trajectories is more than 18 million km and the number of points reaches 8 million. We map each taxi trajectory onto the road network using the map-matching algorithm proposed in [21]. Since more than 25% of road traffic in Beijing is generated by taxis, the taxi trajectory data is a significant sample of traffic flows on the road network of Beijing.

2) Bike Trip Data: We use bike trip data of Citibike sharing system[1] in Manhattan, NYC. The data contain 1,037,712 trips generated by 6,376 bikes and 330 stations in Oct. 2013. Each bike trip has start/stop time, and start/end stations. Bike stations with geographical positions (i.e. vertices) and connections between them (i.e. edges) form an STG G. The inflow and outflow of a bike station can be obtained by counting people returning and renting bikes at the station, respectively. Since New Yorkers make about 113,000 bike trips daily, bike trip data reflect in part traffic flows in NYC. While we use Beijing taxicab data and bike trip data for validation, our method is general enough to adapt other data, such as card swiping data of subway or Call Details Records, as long as they reflect traffic flows on the road network. Figures 7(a) and 7(b) show STGs formed by the two datasets, respectively.



(a) STG of Beijing City (b) STG of Manhattan

Figure 7: STGs Formed by Two Datasets

5.2 Case Study of Beijing City

As shown in Fig. 8, there are 10 black holes and 9 volcanos detected in Beijing City during 14:30 ∼15:00 on Nov. 3, 2012, which located around Beijing Workers' Stadium (denoted by **A**), Beijing South Railway Station (denoted by **B**), Beijing West Railway Station, Sanlitun (shopping center and bar street), Xizhimen (transportation hub) and so on. In the city of Beijing, these regions attract a large amount of people to watch sports games, take trains, shopping and entertain there. People who entered or left these regions by taxis formed the above black holes and volcanos.

5.2.1 Black Holes/Volcanos Representing Anomalies

The irregularly appearing black holes/volcanos represent anomalies in the city. Let us zoom in on Fig. 9, two black holes around Beijing Workers' Stadium represent the anomaly that an important football match between Beijing and Guangzhou Football Teams began at 15:30. In addition, one black hole was formed in 22:00∼22:30, and two volcanos were formed during 22:30∼23:00 on Nov. 24, 2012. A concert in the stadium ended around 22:30 provides us

[1]http://www.citibikenyc.com/system-data/

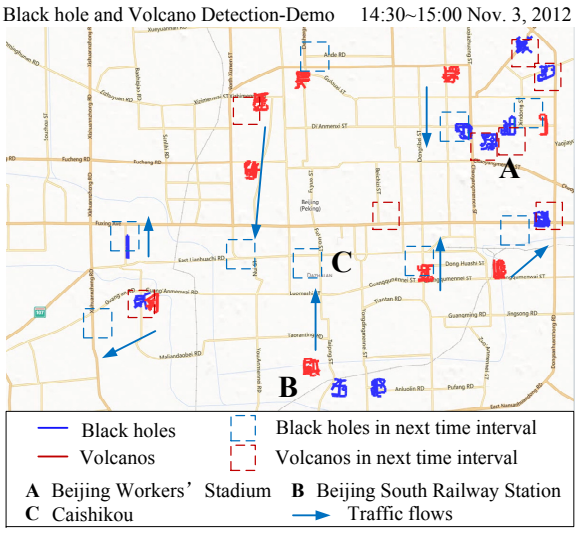


Figure 8: Black holes/Volcanos in Beijing city

with a reasonable explanation. Many taxis came to wait for passengers (i.e. black hole) outside the stadium before the ending, and departed from the stadium (i.e. volcano) after the ending.

The detected anomalies are useful in many applications such as traffic control, event detection and so on. Take the black hole S_1 in 14:30∼15:00 on Nov. 3, 2012 for example. There are 56 edges in S_1. The road network capacity of S_1 is 988 (calculated by the equation in Section 2.1) when the factor α is set to 0.7, which can be used to determine whether the traffic is congested in S_1. Once finding the actual flow of S_1 is approaching the road network capacity 998, the transportation authority can initiate a traffic control to regulate the traffic. The information about the black hole can also be displayed on street-side screens to inform drivers to take alternative routes before a coming traffic jam.

On Nov. 24, 2012, the black hole S_2 in 22:00∼22:30 and the volcano S_3 in the next time interval have 24 overlapping edges. Thus, our continuous detection algorithm can save at least 24 edge accesses by expanding S_2 to S_3.

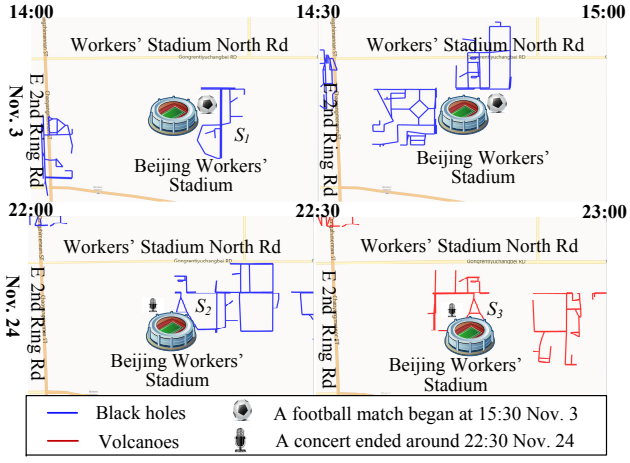


Figure 9: Black holes/Volcanos around Beijing Workers' Stadium

5.2.2 Regular Patterns of Black Holes/Volcanos

The regular black holes/volcanos can be regarded as patterns of black holes/volcanos, which reveal human mobility patterns.

Fig. 10 shows black holes and volcanos around Beijing South Railway Station from 21:30 to 0:00 on Nov. 5 and Nov. 12, 2012, respectively. More than 260 trains depart from or arrive at Bei-

jing South Railway Station every day. Most of them strictly follow the railway time table under normal conditions. Moreover, most of buses and subways are out of service during 21:30 ∼ 0:00. Thus, black holes and volcanos formed by taxis reveal human mobility patterns around the station during 21:30 ∼ 0:00. Four trains were supposed to arrive at the station around 22:00 on Nov. 12. As shown in Figures 10(a) to (d), four volcanos were formed by the departing taxis after 22:00. Such volcanos appear after 22:00 regularly, which become a pattern of volcanos. Note that even though these patterns appear regularly, we still need to detect them continuously, because both the area of influence and duration of black holes/volcanos change dynamically in an STG.

However, as shown in Figures 10 (e) to (h), the pattern delayed for 30 minutes on Nov. 5, 2012. To find out the reason, we search news on Nov. 5, 2012. News articles reported that a heavy snow in North China delayed almost all trains from Shanghai to Beijing by about 30 minutes, which proves the effectiveness of our method.

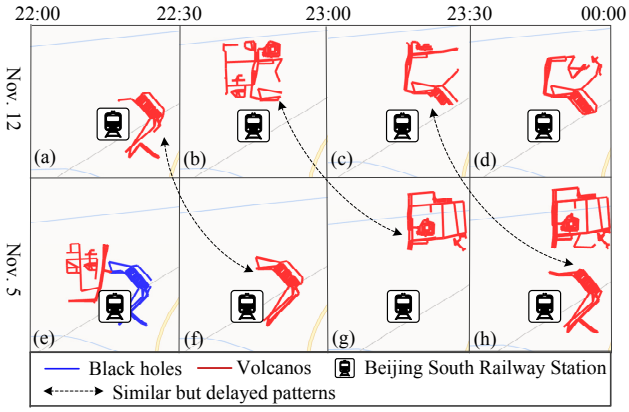


Figure 10: Black Holes/Volcanos at Beijing South Railway Station

Given such pattern, transportation authorities can extend the bus or subway service hour to 22:30 or increase the frequency of buses or subway trains to carry passengers at Beijing South Railway Station around 22:30. In the long run, city planners can expand the capacity of roads around the station to ease the traffic pressure. They can even discover design defects in road network and formulate a better planning of the train station by studying the pattern.

5.2.3 Relations between Black Holes and Volcanos

It is also worth to note that black hole and volcano may transform into each other over time. As shown in Fig. 8, 6 black holes became volcanos, and 3 volcanos became black holes in the next time interval. Specifically, two black holes around Beijing Workers' Stadium became two volcanos in the next time interval. This is because there was a football match between Beijing and Guangzhou football team began at 15:30 on Nov. 3, 2012. Taxis that took football fans to the stadium formed black holes, and then volcanos when leaving the stadium.

Moreover, we find that many people left one volcano for another black hole, i.e. there were heavy traffic flows between volcanos and black holes in two consecutive time intervals. For example, as shown in Fig. 8, 9.8% taxis leave Beijing South Railway Station (i.e. **B**) in the next time interval 15:00∼15:30 for Caishikou (denoted by **C**) which is a famous tourist spot.

The relations between black hole/volcano help better understand city dynamics, which can be used to optimize public transport schedule and city planning. For example, public transport operators can increase the frequency of buses or subway trains between Beijing South Railway Station and Caishikou to help relieve the traffic pressure. City planners can even plan new roads or subway lines that connect Beijing South Railway Station and Caishikou.

5.3 Case Study of New York City

Since people usually rent or return bikes to nearby bike stations, it is necessary to detect black holes/volcanos formed by several bike stations within an area rather than a single station. Fig. 11 shows the black holes and volcanos detected in Manhattan, NYC from 16:00 to 20:00 on Oct. 8 and Oct. 17, 2013. For instance, a volcano located around Wall Street (denoted by **C**) during 17:00 ∼18:00 on Oct. 8, and a black hole located around Time Square (denoted by **A**) during 17:00 ∼18:00 on Oct. 17. Note that, the number of black holes and volcanos is the largest during 17:00 ∼18:00 among all time intervals. This is because traffic flows are at their highest during the rush hour in a work day.

5.3.1 Black Holes/Volcanos Representing Anomalies

As shown in Fig. 11, the volcano around Union Square transformed into a black hole (③) during 17:00∼18:00 on Oct. 17, and two volcanos (⑤) in the next two time intervals. The result is consistent with the event of Union Square Greenmarket from 8:00 to 18:00 on that day. People who rode bikes to leave Union Square after the Greenmarket formed the two volcanos.

Table 1 records the inflow and outflow of bike stations around **A** during 16:00∼20:00 on Oct. 17. For instance, the actual outflow of Station 382 and 497 is 18 and 16 respectively during 16:00∼17:00.

Based on the real time black holes and volcanos, Citibike can temporarily use trucks to deliver bikes to Station 382 and 497 during 16:00∼17:00 and 18:00∼20:00, while move bikes from Station 253, 382 and 497 during 17:00∼18:00. In this case, user experience can be improved by alleviating the shortage of bikes or docks.

5.3.2 Patterns of Black Holes/Volcanos

As shown in Fig. 11, a volcano and a black hole appeared regularly around Union Square (i.e. **A**) during 16:00 ∼ 17:00 and 17:00 ∼ 18:00 respectively on both Oct. 8 and Oct. 17, 2013. The volcano was probably formed by shoppers coming from the 5th Avenue near Union Square which is a famous shopping street, while the black hole was formed by people coming off work from the nearby commercial offices. Similarly, a black hole was formed by employers coming off work from Wall Street (i.e. **C**) during the rush hour 17:00 ∼ 18:00 on both Oct. 8 and Oct. 17. Note that, these patterns are usually beyond the common knowledge of locals.

Based on the above patterns, Citibike can regularly deliver bikies to the stations around Union Square during 16:00∼17:00, while move bikes from these stations during 17:00∼18:00. Citibike can even optimize the design of these stations to alleviate the shortage of docks or bikes. For instance, we should add both docks and bikes to Station 253, 285, 382 and 497, because the actual inflow and outflow of these stations is large in different time intervals as shown in Table 1. However, we only need to add bikes to the stations around Wall Street, which form a volcanos at the rush hour. In the long run, these patterns help Citibike to select sites of bike stations, so that the operational efficiency of the bike sharing system can be improved. Intuitively, Citibike should place more stations at the locations where black holes and volcanos take place frequently.

5.3.3 Relations between Black Holes and Volcanos

As shown in Fig. 11, 3 volcanos transformed into black holes, and 2 black holes transformed into volcanos in total on Oct. 8 and Oct.17, 2013. For instance, the volcano around Grand Central Terminal (i.e. **B**) during 17:00∼18:00 on Oct. 8 transformed into a black hole in the next time interval (②). Such transformation is

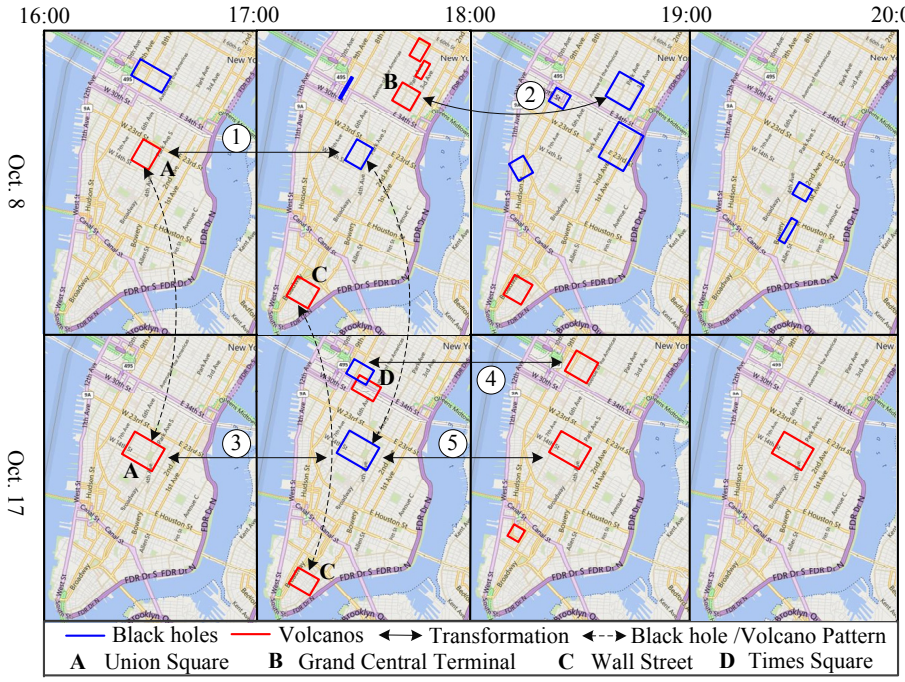


Figure 11: Black holes/Volcanos in New York City

Table 1: Inflow/Outflow of Union Square (A)

Time Interval	Station ID	Inflow	Outflow
16:00-17:00 Oct. 17	382	24	42
	253	15	20
	497	24	40
Volcano	285	12	0
	Total:	**75**	**102**
17:00-18:00 Oct. 17	253	14	9
	382	37	22
	497	56	50
Black Hole	285	19	23
	Total:	**126**	**104**
18:00-19:00 Oct. 17	382	45	63
	253	22	25
	497	53	59
Volcano	285	19	23
	Total:	**139**	**170**
19:00-20:00 Oct. 17	382	30	36
	497	28	32
	253	8	9
Volcano	285	22	21
	Total:	**88**	**98**

probably caused by the cyclists entering or leaving the train station. We also find that many commuters left Wall Street for Grand Central Terminal at rush hours, which is a hot bike route. In addition, the black hole around Times Square (i.e. **D**) during 17:00∼18:00 on Oct. 17 transformed into a volcano during 18:00∼19:00.

The above relations between black holes and volcanos can help NYC planning department to design bicycle lanes (e.g. between Grand Central Terminal and Wall Street) that better relieve traffic congestions in NYC.

5.4 Performance in a Single Time Interval

5.4.1 Baselines

Our method BH_ALL is compared with the following methods:

1) MCL (Markov Cluster) graph clustering algorithm [19] uses stochastic matrix to cluster dense subgraphs based on random walk on the graph. Random walks are calculated using Markov chains. Since MCL is not a continuous algorithm, we repeatedly apply MCL to clustering the STG in each time interval. An STG in a time interval can be regarded as a weighted graph in which flow on each edge represents the weight of the edge. Since MCL does not consider spatio-temporal properties of STG, we check whether each cluster satisfies the spatial threshold and flow threshold after one MCL process, and output the resulting black holes.

2) BL method selects an initial edge e with maximum actual flow in the STG, and then expands e to a black hole by adding randomly selected neighboring edges. The above procedure repeats until no more black holes can be found in the STG.

3) BH_P method uses candidate cell selection together with flow upper bound updating to prune grid cells (denoted by Pruning) , and finds black holes in candidate cells using BL method;

4) BH_E method uses spatial expansion algorithm (denoted by Expansion) to detect black holes in all the grid cells without pruning. The main difference between BH_E and BL is that BH_E method selects neighboring edges according to priority scores while BL method selects neighboring edges randomly.

5) BH_PE method uses candidate cell selection, flow upper bound updating and spatial expansion jointly.

For Beijing taxicab data, the spatial threshold d is set to $\{0.5, \mathbf{1.1}\}\times\sqrt{2}km$, the flow threshold τ is set to $\{20, \mathbf{40}\}$, and the time interval is set to $\{\mathbf{0.5}h, 1h, 1.5h, 2h, 2.5h, 3h\}$. The numbers in bold font are default parameter values. For bike trip data, d is set to $0.4\times\sqrt{2}km$, τ is set to 3, and the time interval is set to $1h$.

All experiments are implemented in C++, and conducted on a dual 3.4GHz Core class machine with 16GB RAM. The operating system is 64-bit Windows 8.

5.4.2 Results

As shown in Table 2, under two parameter settings, BH_ALL outperforms all the competing methods in both running time and average number of detected black holes and volcanos. As more proposed algorithms are applied, more running time can be saved and more black holes and volcanos are detected. Although τ can be determined based on the average road capacity in *MBB* (detailed in Section 2.1), we study the effect of varying τ in this subsection.

Table 2: Running Time and Number of Detected Black Holes and Volcanos in Each Time Interval

Methods	$d=0.5\times\sqrt{2}km, \tau=20$		$d=1.1\times\sqrt{2}km, \tau=40$	
	Time (sec)	Number	Time (sec)	Number
MCL	10.94	0	10.94	0
BL	24.91	3.56 (B) 3.79 (V)	84.95	1.81 (B) 2.16 (V)
BH_E	20.73	14.22 (B) 13.04 (V)	131.49	26.19 (B) 16.07 (V)
BH_P	5.79	3.51 (B) 3.62 (V)	39.29	3.81 (B) 4.1 (V)
BH_PE	4.78	12.44 (B) 11.53 (V)	29.48	22.91 (B) 14.21 (V)
BH_ALL	4.37	19.31 (B) 17.97 (V)	27.18	26.77 (B) 21.20 (V)

First, since spatial expansion algorithm expands an initial edge into a black hole by adding neighboring edges based on priority score, the failure probability of black hole detection will be reduced

and therefore more black holes can be detected. From Table 2, we can see that BH_E detects much more black holes and volcanos than BL method. That is why BH_E takes more time than BL when d and τ increase. Second, candidate cell selection algorithm quickly selects grid cells that could intersect black holes. Moreover, flow upper bound updating algorithm prunes grid cells during spatial expansion. These pruning algorithms reduce the search space of black hole detection, thus greatly save the running time. Table 2 shows that the running time of BH_P is much shorter than that of BH_E. Note that, BH_P detects few black holes and volcanos as it does not employ spatial expansion algorithm. Third, continuous detection algorithm improves both the efficiency and quality of black hole detection by starting from recent black holes/volcanos or black hole patterns. As shown in Table 2, BH_ALL detects at least 16.8% more black holes and 49.2% more volcanos, and saves up to 9.4% running time compared to BH_PE.

As d increases, the running time of all the methods increases as more edges should be accessed with larger spatial threshold. Note that, the running time of BH_ALL increases slowly with regard to the number of detected black holes/volcanos, which proves the scalability of BH_ALL.

Since MCL ignores flow and spatial constraints, its running time does not change with the settings of d and τ. Under both settings, MCL cannot find any black holes, because actual flows of all the clusters found by MCL do not satisfy τ. This is reasonable because MCL ignores spatio-temporal properties of STG, as well as flow and spatial constraints of the black hole in the clustering process.

As shown in Table 3, the total running time 27.18 seconds of BH_ALL is short with regard to the time interval 0.5 hour. Specifically, spatial expansion contributes major part of online running time as the algorithm needs to expand black holes edge-by-edge, while candidate selection and flow upper bound pruning only requires very short calculation time. As d and τ increase, the running time of candidate selection and flow upper bound pruning increases more slowly than that of spatial expansion, because the total number of grid cells in STG index decreases accordingly. Table 3 also reports the running time of map matching.

Table 3: Running Time of Each Component

Setting	Candidate Selection	Spatial Expansion	Flow UB Updating	Map Matching
$d = 0.5\times\sqrt{2}km$ $\tau = 20$	0.21 sec	4.14 sec	0.02 sec	14.82 min
$d = 1.1\times\sqrt{2}km$ $\tau = 40$	0.39 sec	26.72 sec	0.07 sec	

5.5 Performance of Continuous Detection

In this subsection, we evaluate the *reoccurring ratio* and *benefit ratio* of continuous black hole detection. The reoccurring ratio is defined as the percentage of the number of black holes that originate from black hole patterns (denoted by Pattern), recent black holes in the last time interval (denoted by Black Hole), recent volcanos (denoted by Volcano) or any of the above subgraphs (denoted by BH_ALL) to the total number of detected black holes. The reoccurring ratio of volcanos is defined in a similar way, and denoted by Pattern$_V$, Black Hole$_V$, Volcano$_V$, and BH_ALL$_V$, respectively. The benefit ratio is defined as the percentage of average gained time of BH_ALL to the running time of the snapshot method BH_PE.

As shown in Fig. 12, the reoccurring ratios of BH_ALL range from 59% to 74%, which is higher than those of Pattern, Black Hole, and Volcano. The reoccurring ratios of BH_ALL$_V$, Pattern$_V$, Black Hole$_V$, and Volcano$_V$ show a similar trend. As the time interval increases from $0.5h$ to $3h$, reoccurring ratios of all the methods decrease, because less black holes/volcanos reoccur in a longer time interval. Specifically, 27.5%∼53.6% and 32.8%∼43.3% black holes are detected from recent black holes and black hole patterns respectively, 36.9%∼76.3% and 17.5%∼23.4% volcanos are detected from recent volcanos and volcano patterns respectively. Note that, some black holes and volcanos transform themselves into each others: 5.1%∼20.6% volcanos become black holes, and 25.7%∼63.2% volcanos become black holes in the next time interval. The above results prove that continuous black hole detection should utilize black holes and volcanos detected in the past time interval and black hole patterns. Fig. 13 shows the benefit ratio of continuous detection algorithm in a duration of one day (i.e. 24 hours). As shown in Fig. 13, the benefit ratio decreases from 9.1% to 3.3% when the time interval becomes longer. This is because of two reasons. First, given a fixed period of time, longer time interval leads to less number of time intervals and detection results. Second, with longer time interval, the reoccurring ratio decreases.

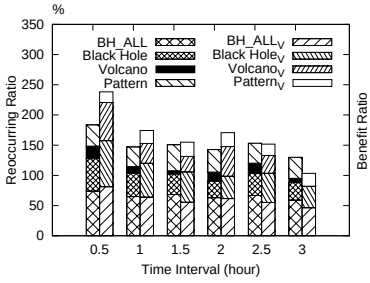


Figure 12: Reoccurring Ratio

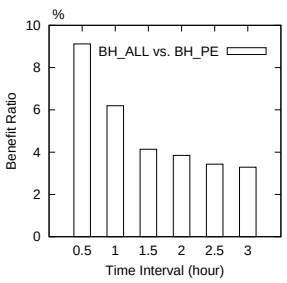


Figure 13: Benefit Ratio

It is difficult to reduce the cost of continuous black hole detection in an STG that evolves with time. As the actual flow changes dynamically, a black hole may not still be black hole in the next time interval, which leads to failures of spatial expansion. Such failure introduces additional cost, and therefore reduces the benefit ratio. In reality, we can use a dynamic lower bound φ of $S.f_a$ rather than 0 in continuous detection algorithm. That is, if the actual flow of the black hole/volcano or black hole pattern (denoted by S) in the past time interval (i.e. $S.f_a$) is lower than φ, we do not expand a black hole from S as the failure probability of spatial expansion would be high. However, dynamic lower bound will reduce the number of detected black holes and volcanos (i.e. the quality of results) because we discard some recent black holes/volcanos and black hole patterns with small actual flow. How to determine φ is a tradeoff between the benefit ratio and the quality of results, which can be set by specific applications according to real requirements.

6. RELATED WORK

Graph Clustering: MCL [19] is a graph clustering algorithm using stochastic matrix. MLR-MCL [16] is a multi-level regularized graph clustering algorithm that finds clusters on a coarsen graph based on MCL. Li et al. [9, 10] formulated the problem of mining black hole and volcano patterns (i.e. clusters) in a large directed graph, and proposed two pruning schemes to guide the mining process. There are also other variants of graph clustering based on different measures for identifying clusters, such as attributed graph clustering [25], modularity-based graph clustering [1], density-based graph clustering [17] and so on. Our work differs from the above graph mining problems in that it employs a different cluster measure (i.e. flow and spatio-temporal constraints) to detect black holes/volcanos in an STG instead of a static graph.

Time-evolving Graph Mining: Liu et al. [11] proposed an incremental algorithm to efficiently mine significant subgraphs in evolv-

ing graphs. A constraint-based pattern mining approach was proposed to find pseudo-cliques in dynamic graphs [15]. In [6], Lahiri et al. proposed an efficient pattern-tree-based algorithm to mine periodically recurring interaction patterns in dynamic networks. A recent work [14] deals with query processing over historical evolving graph sequences to obtain interesting information from various query results. Differently, since spatio-temporal graph has spatial properties, our problem should consider spatio-temporal constraints in black hole and volcano detection.

Grouping Traveling Behavior: The problem of density querying for moving objects finds out regions that have objects higher than a threshold in a given time interval [3]. Jensen et al. [4] generalized the problem to a dense region query with fixed shape. The differences between density querying [3, 4] and our work are: 1) A dense region is not necessary to be a black hole or volcano, and vice versa. For example, a dense region is not a black hole if all the moving objects finally move out of the regions within the time interval. 2) The irregular shape black holes/volcanos cannot be found by [3, 4] even if we change the definition of density from high concentration of objects to high actual flow. Since [3, 4] consider fixed shape region as a whole, these methods fail to detect black holes with irregular shapes because a fixed shape region may include other edges with negative actual flows than edges within the irregular shape region. Another branch of research is to discover a group of objects that move together for a certain time period (i.e. trajectory clustering), such as convoy [5], swarm [8], traveling companion [18] and gathering [22]. However, black holes are not necessarily formed by a group of similar trajectories.

Spatio-Temporal Data Mining: Mathioudakis et al. [12] introduced scalable algorithms to identify spatial information bursts. Lappas et al. [7] presented a framework for simultaneously tracking the spatial and temporal burstiness of terms. Note that, our problem is more challenging than the problem of identifying spatio-temporal bursts which ignore graph topology of spatial regions.

Recently, Pan et al. [13] studied the problem of detecting and describing traffic anomalies using crowd sensing with human mobility and social media data. Chen et al. [2] developed nonparametric learning methods to learn dynamic graph structures from spatial-temporal data. There is plenty of literature on trajectory pattern mining [23], aiming to analyze the mobility patterns of moving objects. The problem definition of our work is different from prior works which only consider the total amount of flow while ignore the inflow and outflow of black holes/volcanos.

7. CONCLUSIONS

In this paper, we model human mobility data by an STG, from which we detect urban black holes. Case study on Beijing taxicab data demonstrates that our method can instantly find urban anomalies (e.g. football matches and concerts in Beijing Worker's Stadium) that faciliate early warning of traffic congestions and temporary traffic control, and human mobility patterns (e.g. the patterns around Beijing South Railway Station) that help improve the urban planning of Beijing. Case study on Bike trip data shows that the instantly detected black holes and volcanos can assist real-time scheduling of bikes, and the black hole/volcano patterns help optimize the deployment and site selection of bike stations. Moreover, our method outperforms baseline methods by reducing at least 68% running time and detecting 10 times more black holes. Compared to the black hole detection in a time interval, our continuous detection algorithm saves more than 9% total computational cost.

In the future, we would like to apply our method to Call Details Records. The detected black holes would indicate the emerging events, helping avoid potential risks of public safety.

Acknowledgments

The research is supported by NSFC under Grant 61303025, 61572488, 863 project under grant 2015AA015402, NSFC (Key Program) under grant 61532010, Beijing Higher Education Young Elite Teacher Project (YETP0016), and the Fundamental Research Funds for the Central Universities.

8. REFERENCES

[1] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Trans. on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[2] X. Chen, Y. Liu, H. Liu, and J. G. Carbonell. Learning spatial -temporal varying graphs with applications to climate data analysis. In *Proc. of AAAI*, 2010.

[3] M. Hadjieleftheriou, G. Kollios, D. Gunopulos, and V. J. Tsotras. On-line discovery of dense areas in spatio-temporal databses. In *Proc.of SSTD*, 2003.

[4] C. S. Jensen, D. Lin, B. C. Ooi, and R. Zhang. Effective density queries on continuously moving objects. In *Proc. of ICDE*, 2006.

[5] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of convoys in trajectory databases. *PVLDB*, 1(1):1068–1080, 2008.

[6] M. Lahiri and T. Y. Berger-Wolf. Periodic subgraph mining in dynamic networks. *Knowledge and Information Systems*, 24(3), 2010.

[7] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras. On the spatiotemporal burstiness of terms. *PVLDB*, 5(9):836–847, 2012.

[8] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *PVLDB*, 3(1-2):723–734, 2010.

[9] Z. Li, H. Xiong, and Y. Liu. Mining blackhole and volcano patterns in directed graphs: A general approach. *Data Mining and Knowledge Discovery*, 2012(25), 2012.

[10] Z. Li, H. Xiong, Y. Liu, and A. Zhou. Detecting blackhole and volcano patterns in directed networks. In *Proc. of ICDM*, 2010.

[11] Z. Liu, J. X. Yu, Y. Ke, and X. Lin. Spotting significant changing subgraphs in evolving graphs. In *Proc. of ICDM*, 2008.

[12] M. Mathioudakis, N. Bansal, and N. Koudas. Identifying, attributing and describing spatial bursts. *PVLDB*, 3(1-2):1091–1102, 2010.

[13] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi. Crowd sensing of traffic anomalies based on human mobility and social media. In *Proc.of GIS*, 2013.

[14] C. Ren, E. Lo, B. Kao, X. Zhu, and R. Cheng. On querying historical evolving graph sequences. In *PVLDB*, 2011.

[15] C. Robardet. Constraint-based pattern mining in dynamic graphs. In *Proc. of ICDM*, 2009.

[16] V. Satuluri and S. Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *Proc. of KDD*, pages 737–746. ACM, 2009.

[17] J. Šíma and S. E. Schaeffer. On the np-completeness of some graph cluster measures. In *Proc. of SOFSEM*, pages 530–537. Springer, 2006.

[18] L.-A. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, C.-C. Hung, and W.-C. Peng. On discovery of traveling companions from streaming trajectories. In *Proc. of ICDE*, 2012.

[19] S. Van Dongen. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, 2008.

[20] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proc. of ICDM*, 2002.

[21] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Sun. An interactive -voting based map matching algorithm. In *Proc. of MDM*, 2010.

[22] K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang. On discovery of gathering patterns from trajectories. In *Proc. of ICDE*, 2013.

[23] Y. Zheng. Trajectory data mining: An overview. *ACM Trans. on Intelligent Systems and Technology*, 2015.

[24] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: concepts, methodologies, and applications. *ACM Trans. on Intelligent Systems and Technology*, 2014.

[25] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. In *Proc. of VLDB Conference*, 2009.