Inferring the Root Cause in Road Traffic Anomalies

Sanjay Chawla*, Yu Zheng† and Jiafeng Hu‡
* *School of Information Technologies*
University of Sydney, Sydney, Australia
Email: sanjay.chawla@sydney.edu.au
† *Microsoft Research Asia, Beijing, China*
Email: yuzheng@microsoft.com
‡ *Institute of Software, Chinese Academy of Sciences, Beijing, China*
Email: acmhujiafeng@gmail.com

***Abstract*—We propose a novel two-step mining and optimization framework for inferring the root cause of anomalies that appear in road traffic data. We model road traffic as a time-dependent flow on a network formed by partitioning a city into regions bounded by major roads. In the first step we identify link anomalies based on their deviation from their historical traffic profile. However, link anomalies on their own shed very little light on what *caused* them to be anomalous. In the second step we take a *generative* approach by modeling the flow in a network in terms of the origin-destination (OD) matrix which physically relates the latent flow between origin and destination and the observable flow on the links. The key insight is that instead of using all of link traffic as the observable vector we only use the link anomaly vector. By solving an L_1 inverse problem we infer the routes (the origin-destination pairs) which gave rise to the link anomalies. Experiments on a very large GPS data set consisting on nearly eight hundred million data points demonstrate that we can discover routes which can clearly explain the appearance of link anomalies. The use of optimization techniques to explain observable anomalies in a generative fashion is, to the best of our knowledge, entirely novel.**

I. Introduction

The flow of traffic on a road network is a complex phenomenon. A small event can cause a dramatic change in the flow and can propagate in an uneven manner throughout the system. The challenge from a data mining perspective is that a historical archive of traffic flow usually does not explicitly contain a description of events which may have caused perturbations in the system. While existing data mining techniques (especially anomaly detection) can be applied to mine for deviations, there is no known systematic way to piece together the mined anomalies to infer events which may have caused the anomalies to occur.

To give a concrete example, consider the setting shown in Figure 1 about Beijing's road network. Our objective was to find interesting patterns from GPS data obtained from Beijing's taxi cabs. We first applied PCA to search for anomalous links connecting two regions, based on their historical pattern. An example of a discovered anomaly is shown as a red (bold) arrow. On its own it is difficult to explain why the discovered anomaly would be interesting.

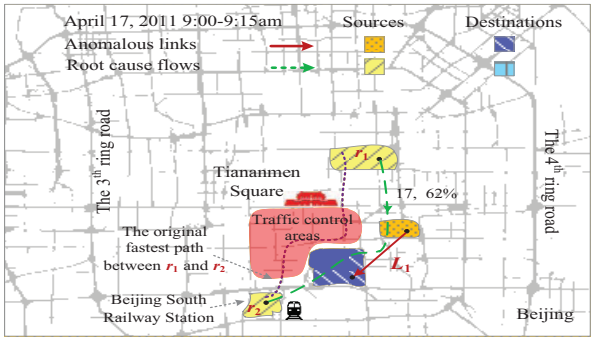


Figure 1. Bold (red) links were discovered using Principal Component Analysis (PCA) techniques for anomaly detection. Relationship between links and routes was captured using the link-route matrix. Routes were inferred using L_1 optimization techniques. It became easier to explain the route anomaly because of the re-routing of traffic due to the Beijing marathon. This overall increases the precision of the discovery process.

However, in the next step we combined links with routes using a link-route matrix and then used L_1 optimization techniques to infer routes which may have caused the anomalous links to appear. This is shown as a green (dashed) path. The discovery of the anomalous route helps put the data mining exercise in perspective and explain the anomaly. As it turned out, on April 17th, 2011, traffic in Beijing had been diverted away from Tiananmen Square because of the Beijing marathon. Thus the normal traffic route (shown as dotted path) from region r_1 to the Beijing South Railway Station was diverted and the dashed (green) path witnessed excess traffic. The automatic discovery of anomalous routes caused by diversion in traffic can now be used for future planning by the city road authority. The relationship between the anomalous links and the routes (which may have caused them), helps increase the precision of the discovery process. We will give several more real examples in the Experiment and Evaluation section.

We make the following contributions:

1) We propose a novel two-step mining and optimization framework to infer events which may have caused anomalous behaviour to appear in road traffic flow.

2) We model and study the traffic between regions rather than on road surfaces. This not only reduces the complexity of the model but also helps with the detection of the root cause of traffic anomalies.
3) We validate our framework using a large GPS trace consisting of nearly eight hundred million data points. Using our approach we were able to infer real events which caused perturbations in the traffic flow. Information about these events was not part of the original data set.

The rest of the paper is organized as follows. In Section 2 we define the problem and set up the notation. In Section 2 we explain our methodology which combines the use of anomaly detection with optimization techniques. The experimental setup and evaluation is described in Section 4. We overview related work in Section 5. We conclude in Section 6 with a summary and directions for future work.

II. Traffic Modeling

We model a road network as a directed graph $\mathcal{N} = (\mathcal{V}, \mathcal{L}))$ where V is the set of regions bounded by major roads and L is the set of directed links that connect two regions. For now we will assume that both V and L are fixed but later we will see that the set L can change as a function of time.

As demonstrated in Figure 2(A), the region map of Beijing is partitioned by major roads. Each region is modeled as a node of a graph [16]. To define the links we first observe the flow of taxis and based on parameters (defined later), connect two regions with a link if sufficient taxi flow exists between the two regions for a given time window. Example flows are shown in (B). Based on the flows we define routes or paths between regions. For example in (C), paths which end in region r_4 are shown. The abstract graph which captures regions and links between them is shown in (D). The decision to model regions (rather than say traffic intersections) was based on two considerations. The first is that regions (bounded by major roads) have a semantic coherence. For example a region could represent a business zone, shopping district, a cluster of a higher education entities or residential locations. Each of these semantic zones have their unique mobility patterns. For example, if a link connecting a residential area to a shopping district shows abnormally high traffic compared to usual then that is an indicator that the anomaly is probably due to a holiday. We have found several such anomalies and they will be reported in the Experiment section. The second reason for modeling regions is efficiency and handling data inaccuracies. The region graph is substantially smaller and furthermore the inaccuracies of GPS sensors can be averaged out when we deal with larger regions. More details about the segmentation algorithm to form regions and inferring the semantic function of regions can be found in [16].

Term	Notation	Descrption
Link-Route Matrix	$\boldsymbol{A}$	$\{0-1\}$ binary matrix
Link Time Matrix	$\boldsymbol{L}$	Real-valued matrix
The link anomaly vector	$\boldsymbol{b}$	$\{0,1\}$ vector
PCA eigenvalues	λ_i's	non-zero
PCA eigenvector	$\boldsymbol{v_i}$'s	real-valued
L_1 norm	$\|\boldsymbol{x}\|_1$	$\|x_1\| + \ldots \|x_n\|$
L_0 norm	$\|\boldsymbol{x}\|_0$	non-zero $\|x_i\|$'s
L_2 norm	$\|\boldsymbol{x}\|_2$	$\|x_1\|^2 + \ldots \|x_n\|^2$
The route vector	$\boldsymbol{x}$	$\{0,1\}$ valued from $\boldsymbol{Ax} = \boldsymbol{b}$.
Path element	$\boldsymbol{p_i}$	a path connects o-d pair

Table I
IMPORTANT NOTATION THAT WILL BE USED IN THE PAPER

We capture the relationship between links and the routes (paths) as a link-route binary matrix A. The entries of the link-route matrix are given by

$$A_{ij} = \begin{cases} 1 & \text{if link } i \text{ is on route } j \\ 0 & \text{otherwise} \end{cases}$$

An example link-route matrix is shown in (E). The distinguishing feature of the link-route matrix is that the number of possible routes (n) is typically much greater than the number of links (m). The traffic flow on a particular link is a function of all the traffic that flows on routes that contain that link. Thus if we associate a link flow vector b which contains flow information of the traffic of links and x as the flow vector of routes then under equilibrium conditions we can model the relationship between the route flow vector x and the link flow vector b as

$$Ax = b \tag{1}$$

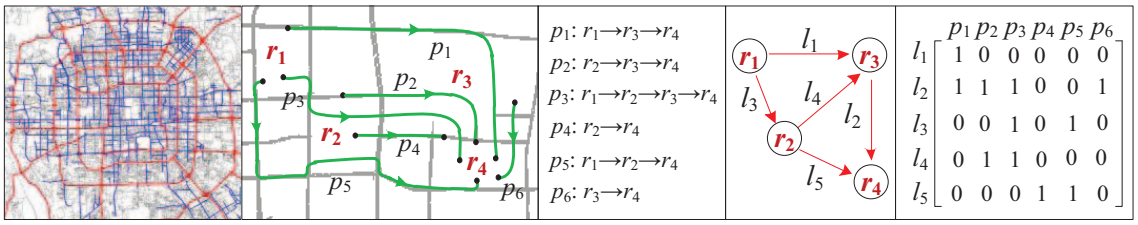
Using GPS technology we can monitor the flow of traffic on links in a given time period. For example, the following is an example of a link traffic matrix L across five time periods:

	$t1$	$t2$	$t3$	$t4$	$t5$
l_1	10	20	10	20	10
l_2	5	5	5	5	5
l_3	20	10	50	70	80
l_4	10	50	60	20	10
l_5	12	20	30	40	50

The link matrix L and the adjacency matrix A will play a crucial role in subsequent analysis. In our proposed two step mining and optimization approach, we will first apply PCA to mine for link anomalies from L. We will then apply L_1 optimization techniques on $Ax = b$ to infer possible routes that may have caused the link anomalies. Table I lists the important notation that is used throughout the paper.

III. Methodology

In this section we describe in detail the components of our methodology to infer routes which may have *caused* the link anomalies. In Section III-A we will describe the use of Principal Component Analysis (PCA) to detect anomalies



A) A region map B) Flows (by routes) C) Paths D) Links (region graph) E) Link-route matrix

Figure 2. An example using traffic networks in the city of Beijing. Based on major roads in the traffic network, the entire city (subfigure (a)) is partitioned into regions (subfigure (b)). Trajectories of moving objects (such as a moving taxi shown by a blue trajectory in subfigure (c)) connect neighboring regions, based on which we create the notion of links (subfigure (d)).

from the link traffic matrix L. In Section III-B1 we will explain the rationale behind the use of L_1 minimization to infer information about routes why may have caused link anomalies.

A. PCA for Anomaly Detection

In this section we describe the use for Principal Component Analysis (PCA) for anomaly detection.

PCA is a widely used dimensionality reduction and lossy compression technique in Data Mining [15], [10]. PCA exploits the observation that in most explicitly high-dimensional data sets, there is a high implicit correlation between many dimensions (variables) which can be inferred by carrying out an eigen-decomposition of the data covariance matrix. Before we explain the method is detail we provide the high level idea behind the use of PCA.

PCA selects a new data-dependent basis for the data. These basis are called the principal components. The first basis element is in the direction of maximum variance, the second in the direction of second highest variance and so on. The principal components are the eigenvectors of the covariance matrix of the data which is always symmetric and semi-positive definite. It has been observed that for most real data sets much of the variance will reside along a small percentage of higher principal components (i.e., those corresponding to higher eigenvalues). Thus by projecting the data into the first few principal components, most of the variance in the data can be preserved while simultaneously reducing the dimensionality.

Now the basic intuition behind the use of PCA for anomaly detection is that for a normal data point most of its norm is concentrated in the subspace spanned by the higher principal components. *Contrapositively, if for a data point most of its norm is concentrated in the subspace spanned by the lower principal components then it is a candidate anomaly!*

Deciding the split which separates the higher eigenvalues from the lower eigenvalues is often data dependent and is one of the weaknesses of the PCA method. A conventional approach is to use the eigenvectors of the top-k eigenvalues for the normal subspace, where top-k captures around 95% of the variance in the sample data.

The advantage of PCA is that both spatial and temporal correlation can be captured by specifying the covariance matrix structure appropriately. The disadvantage of PCA is that separating the "normal" subspace from the "abnormal" subspace is often arbitrary and the results are sensitive to the choice made.

Consider the L matrix which consists of the evolutions of link traffic over time. Here the rows are the links and the columns are the time bins. The following steps need to be carried out to determine candidate anomalies.

1) Let $\tilde{L} = L - \mu$, where μ is the column sample mean matrix.
2) Form the matrix $C = \tilde{L}^T \tilde{L}$. C is an $t \times t$ matrix where t is the number of time intervals being used. For example we could restrict the time to a few hours or the full day depending upon the granularity of the analysis required. Note that our choice of C is determined by the fact that we will be searching for link anomalies rather than time intervals which are anomalous (as is the case in the networking community).
3) Compute the eigendecomposition of C, i.e., all eigenvalue-eigenvector pairs (λ_i, v_i), such that
$$Cv_i = \lambda_i v_i$$
4) Order the pairs (λ_i, v_i) in decreasing order of eigenvalues
$$\lambda_1 \geq \lambda_2 \geq \ldots, \lambda_k, \lambda_{k+1} = \lambda_{k+2} = \ldots \lambda_t = 0$$
5) Let P_n be the subspace $[v_1, \ldots, v_r]$ of R^t spanned by the first r eigenvectors. Similarly P_a be the subspace spanned by $[v_{r+1}, \ldots, v_t]$.
6) Project all data points onto P_a. Thus if x is a original data point then denote by x_a its projection in P_a
7) Define a threshold θ and select all links for which $\|x - x_a\| > \theta$ as candidate anomalies.

Thus there are two important parameters which can have a strong bearing on the selection of candidate anomalies. The first is the choice of eigenvalue λ_r which will determine the

formation of the normal and abnormal subspace P_n and P_a. The second parameter is the choice of θ to select candidate anomalies.

1) Choice of Covariance Matrix: The choice of the covariance matrix has an important bearing on the type of correlations that are being captured by PCA. For example, for the link matrix L, LL^T captures the spatial correlation between the links, while L^TL will capture the temporal correlation. We can also capture spatio-temporal correlation by using the Karhune-Lowe transform. An interesting discussion on how the choice of the covariance matrix can effect the subsequent analysis and interpretation can be found in [2].

2) Example of PCA Anomaly Detection: We present a small example to illustrate the use of PCA for anomaly detection. Consider the 5×5 link matrix shown in Section 2. For observation it is clear that l_4 exhibits anomalous behaviour as the traffic counts in time steps four and five suddenly drops compared to its past counts and also vis-vis the behaviour of other links.

To carry out a PCA analysis we first normalize the L matrix and form the 5×5 L^TL covariance matrix. An eigendecomposition of the covariance matrix show that the eigenvalues in decreasing order are

$$[1.9 \times 10^3, 0.67 \times 10^3, 0.02 \times 10^3, 0.01 \times 10^3, 0]$$

We choose the first eigenvector as the normal subspace P_n and the remaining eigenvectors as the abnormal subspace P_a. All the points are projected onto P_a and in this space for all points we compute the square of the deviation from the mean. These are

$$[0.4 \times 10^3, 0.06 \times 10^3, 0.5 \times 10^3, \mathbf{1.47 \times 10^3}, 0.49 \times 10^3]$$

Thus the technique correctly identifies link 4 as the anomaly.

B. Inferring Routes from Anomalous Links

Have discovered the anomalous links using PCA as described in the previous section, we now describe how we can infer routes whose flow traffic may have caused the anomalies.

The problem of inferencing origin-destination pairs and routes from link traffic data has been intensively studied both in the transportation and the networking (Internet) community. In transportation research this problem is sometimes known as the *observability* problem and in networking research it has often referred to as the network tomography problem [5], [17]. There are two characteristics of this problem that we highlight.

1) In equilibrium, the relationship between traffic flowing on a route and links on the route is given by a simple linear relationship
$$\boldsymbol{Ax = b}$$
Here A is the $\{0, 1\}$ link-route adjacency matrix, x is the route vector of traffic flows and b is the vector of flows on the links. To reiterate, this is an *idealized* relationship which is hypothetically assumed to hold in equilibrium. In practice there is time-dependency between the origin-destination and link flows.
2) The system of equation $\boldsymbol{Ax = b}$ is under-constrained. This is because the number of possible routes is substantially greater than the number of links. This implies that by itself there are infinitely many solutions to the system of equations.

1) L_0 and L_1 Solutions: The problem that an under-constrained system will result in infinitely many solutions can be addressed by specifying the type of the solution that is required by the application. For example, we can require the returned solution to have small component values or be sparse. We note that much of the recent interest in sparse solution for systems of equations and compressed sensing address exactly the issue that we will highlight [4].

A natural way to enforce sparsity is to use the L_0 norm which is defined as

$$\|x\|_0 = |\{x_i | x_i \neq 0\}|$$

One of the surprising results that has received prominent attention lately is that if the L_0 norm is replaced with the convex L_1 norm, $\|x\|_1 = \sum_i |x_i|$, then the solution returned can still be sparse.

To get an insight on why the L_1 and L_0 solution may coincide we consider the simple case of the system

$$\min \|x\|_0 \text{ s.t.} a_1x_1 + a_2x_2 = b_1$$

We can convert the L_1 relaxation of the above equation into a Linear Programming(LP) formulation. For ease of exposition assume all components of the problem are non-negative. By introducing an additional variable t the above minimization problem can be expressed as

$$\min \ t \tag{2}$$
$$a_1x_1 + a_2x_2 = b_1 \tag{3}$$
$$x_1 + x_2 = t \tag{4}$$

The LP is depicted in Figure 3. Now, since we are minimizing t and want to satisfy the constraint at the same time, the hyperplane $x_1 + x_2 = t$ moves towards the fixed constraint and stops the moment the constraint is satisfied. Thus if $a_1 > a_2$, the moving hyperplane will touch the constraint at the point $(0, b_1/a_1)$ which is exactly one of the $lL0$ solutions. Furthermore notice that for the L_1 solution to be sparse, the two constraints could not have been parallel (or linearly dependent). In fact this turns out to be a crucial condition to guarantee the sparseness of the L_1 solution [3]

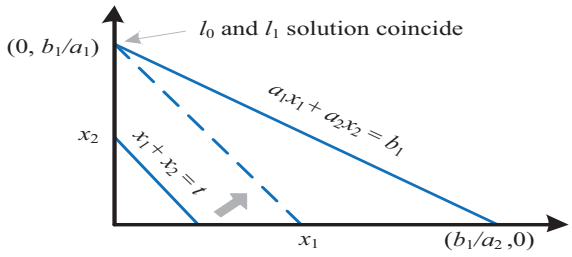


Figure 3. Explanation of when the L_1 and L_0 solution coincide. If the L_1 problem is formulated as an LP then the hyperplane $x_1 + x_2 = t$ moves the least possible amount until it first touches the hyperplane constraint.

2) Discussion on L_1 solution: As we will demonstrate in the Experiment section, the L_1 solution plays a key role in selecting routes which may have caused the emergence of anomalous links. The space of routes has a much greater cardinality than the space of links. However, the L_1 solution, by being sparse, prevents the number of possible candidates from exploding. Yet at the same time, information about routes is much easier to interpret.

3) Example: L_1 solution: **Example:** Using the link-path matrix A given in Figure 2, and suppose link l_2 and link l_4 are anomalies. Then $b = [0, 1, 0, 1, 0]^T$ is the link anomaly vector. The dimensionality of A is a 5×6. The matrix A is

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

The number of possible routes is six and denote the unknown vector of routes by x. We apply the well known Matlab convex solver **cvx**[8] to obtain a solution of the L_1 constrained optimization problem and receive

$$x = (0, 1, 0, 0, 0, 0)$$

This solution is semantically meaningful as it indicates that the following path have a bearing on the link anomalies

$$p_2 : r_2 \rightarrow r_3 \rightarrow r_4$$

Now if we solve for the l_2 solution we receive

$$x = (0, 0.75, 0.25, 0.25, -0.25, 0.0)$$

This makes it very hard to interpret which routes are related to the link anomalies.

IV. Experiment and Evaluation

In this section, we evaluate both efficiency and effectiveness of our methods using real-world trajectories obtained from GPS-equipped taxicabs in Beijing. These cabs can be regarded as mobile sensors constantly probing the traffic flow on road surfaces. Our approach is implemented on a 64-bit server running Windows Server 2008 (OS) with a 2.66GHZ CPU and a 16G memory.

A. Setting

Taxi Trajectories: We use GPS trajectories generated by 13,597 taxis over a period of 3 months (March, May, and August in 2011). The total distance of the dataset is over 400 million kilometers and the total number of GPS point is almost 790 million. The average sampling interval of the dataset is 70.4 seconds. From the taxi trajectories we identify effective trips (the taxicab was carrying a passenger)from the an embedded weight sensor. As a result, 8,202,012 trips have been detected, which is over 15 percent of traffic on road surface (according to the report of Beijing Transportation Bureau).

Road Network: The road network of Beijing consists of 121,771 road nodes and 162,246 edges. Using the major roads (there is a road level associated with each edge) from the network, Beijing has been partitioned into 580 regions. As illustrated in Figure 4 we define 15 minutes as a time interval and study the performance of our method changing over the size of the sliding window w. That is, we carry out our method every 15 minutes using the taxi trajectories received in the past w hours. The length of a time interval is a trade-off between the computational load and the timeliness of an application. On the one hand, setting a long time interval reduces the times of anomaly detection but will lead to a slow notification once an anomaly occurs. On the other hand, a too short time interval (like 5 minutes) will waste unnecessary computing resources as the traffic flow will not change too much in a short period. We study the performance of our method changing over window size w.

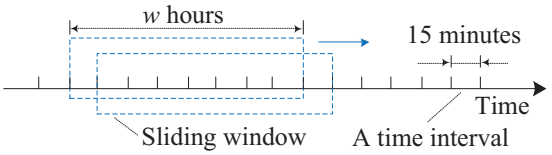


Figure 4. The sliding window and matrix update time

In each implementation, we map the taxi trajectories received in the latest time interval onto the road network, updating the link-route matrix. For example, Figure 6 shows the number of trajectories, paths, OD pairs, and links (original as well as after being filtered) of a weekday (5/18/2011) and weekend (4/17/2011), using 2 hours as the windows size. In this case, we filter some links traversed by less than 5 trajectories in every time interval of the past two hours. These links may have been caused by noisy trajectories or due to the imperfectness of the map-matching function. Additionally, in practice, we only need to capture the significant anomalies instead of all. In the later

experiments, we found that the performance of our method is not compromised by using the small set of links. Further, Figure 5 plots the trajectory data (the lighter the denser) and link graph generated in 5/18/2011 on Beijing map.

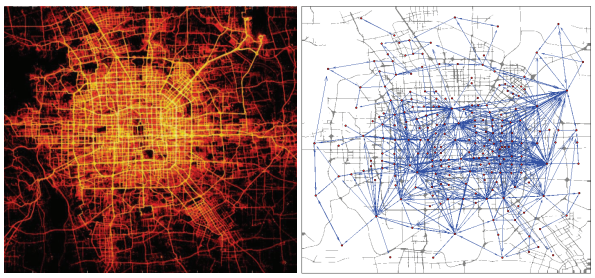


Figure 5. Trajectory data and link distributions on maps

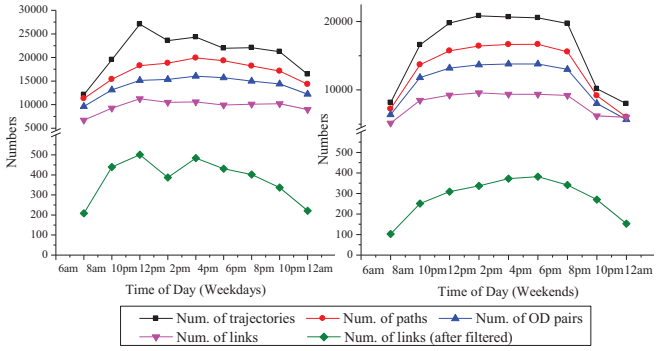


Figure 6. Traffic and links changing over time of day

B. *Efficiency*

Table II shows the efficiency of our framework, using the average time cost of each component. For PCA we used the standard inbuilt algorithm in Matlab and for L_1 we used the cvx package [8]. We also implemented a greedy version of L_1 similar in spirit to the Basis Pursuit algorithm [6]. The building of matrices like A and L was programmed in C#. Note that once these matrices have been built for the first time, we can update them very quickly in the streaming scenario. Clearly, our method can be carried out very efficiently for online applications. In short, in most instances our framework can detect anomalies within 10 seconds. In addition, extending the window size only leads to a slight increase to the computing time. Given a sliding window of 8 hours, we can still find anomalies within 15 seconds. These results demonstrate the efficiency and scalability of our method. The efficiency can be further enhanced using some updating strategies proposed in streaming databases.

We studied three algorithms, consisting of cvx-L_1, cvx-L_2, and the L_1-greedy, that can be selected in the second step of our framework. As shown in Figure 7 A), l_2 algorithm has the best efficiency according to the mean running time.

W	Building matrices		PCA	L_1-cvx	Total
(h)	First time(s)	Update(s)	(s)	(s)	(s)
1	16	1.8	0.03	0.75	2.58
2	20	2.4	0.04	1.37	3.80
3	42	2.8	0.04	2.05	4.89
4	55	3.2	0.07	2.81	6.07
5	63	4.1	0.07	3.57	7.44
6	65	4.1	0.08	4.43	8.61
7	74	4.5	0.08	5.39	10.01
8	82	4.9	0.08	6.44	11.48

Table II
THE BUILD, UPDATE AND RUNNING TIME OF MATRICES, PCA AND L_1 OPTIMIZATION RESPECTIVELY

However, we found that l_2 will result in many non-zero entries in the vector x (refer to Table 2). That is finding many routes contributing to an anomalous link, thereby making it harder to interpret the results. As demonstrated in Figure 7 B), the L_1-greedy algorithm is faster than cvx-L_1 when the size of the sliding window is small. As the window size increases, cvx-L_1 demonstrates its advantages over the L_1-greedy algorithm. So, we can choose different L_1 implementation when using different window sizes.

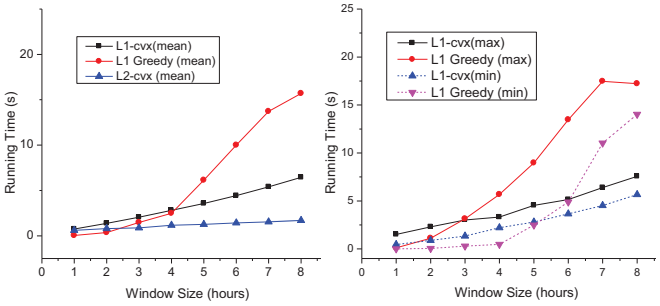


Figure 7. Efficiency of L_1 implementation

Figure 8 shows the distribution of the distance of the links to the mean value in the anomalous space (after being transformed by PCA). No matter what size of a window and what time of day with which we studied the distribution, 90% of the links have a distance smaller than one times the standard deviation to the mean value and 98% had a distance less than three standard deviations. Based on this analysis, links whose distance to the mean was greater than three standard deviations were labeled as anomalous.

C. *Effectiveness*

We evaluated the effectiveness our solution using both real case studies and semi-synthetic experiments. Table III presents the average number of detected anomalous links and number of paths contributing to these links using different window size. Generally, a larger sliding window leads to more anomalous links and paths (the paths only traversed by one taxi trajectory have been filtered). Unlike the L_2

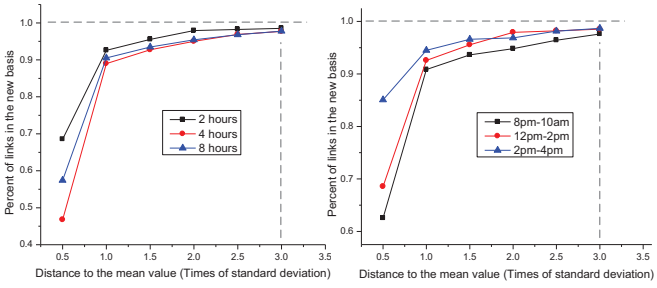


Figure 8. Distribution of deviations in the anomalous subspace

algorithm that returns hundreds of paths, cvx-L_1 offers a reasonable number of paths when analyzing the root cause of the detected anomalous links.

W (h)	Ave. num of anomalous links	Ave. number of non-zero entries of x		
		L_1-cvx	L_1-greedy	L_2
1	4	10	4	40
2	5	23	14	88
3	9	30	31	192
4	10	31	48	250
5	13	58	88	410
6	14	63	107	524
7	16	97	114	650
8	17	91	100	693

Table III
NUMBER OF DETECTED ANOMALOUS LINKS AND PATHS. IT IS CLEAR THAT THE L_1 SOLUTION IS SUBSTANTIALLY MORE SPARSER THAN L_2. FURTHEMORE, EVEN THOUGH MORE ROUTES THAN LINKS ARE RETURNED, ROUTE ANOMALIES ARE EASIER TO EXPLAIN.

1) Real Case Studies: We further evaluated the detected anomalies based on real-world events reported by Beijing Transportation Bureau (as it is difficult to obtain all the ground truth for the detected results). Figure 9 highlights some events that occurred on a workday and non-workday respectively, using a 2-hour window size and 15-minute time interval. Figure 9(I) A) depicts some anomalies we detected on (9am-11am) 4/2/2011 which should be a weekend but was rescheduled to be a workday due to the upcoming Tomb-Sweeping Festival (4/3-4/5). This is also the first weekend after the opening of Sakura Festival held in Yuyuantan Park which is located in region r_3. Though most people went to work, a large amount of traffic was still been generated by people who traveled (especially from r_1, r_2, and r_4) to r_3 to participate in the Sakura festival. As a result, three anomalous links ($L1$, $L2$, and $L3$) were detected. Note that our framework does not only identify anomalies but also find out the root cause traffic leading to the anomalies. For example, 21.4% of the traffic passing $L2$ was from r_1 and 14.3% was from r_2. At the same time, 61% of the traffic causing $L3$ originated in r_3. We further present the traffic volume on $L1$ and $L2$ in Figure 9(I)B) and C) where we can see the sudden changes (marked with red circle) of traffic on these two links from 9am to 11am. Knowing that regions r_1, r_2, and r_4 are regions of historic interests (containing many tourist attractions), we obtained the whole picture about these events. That is, these anomalies could be generated by tourists who want to attend the Sakura Festival.

As presented in Figure 9(II), a traffic control was enforced in the red area for a Marathon race starting from Tiananmen Square, leading to the blockage of the fastest routes between region r_1 and r_3, r_2 and r_3, r_1 and r_5. An anomaly occurred on link $L1$ due to the decrease of traffic flows as shown in Figure 9(II)-B). In other words, people from r_1, r_2, and r_6 have to take a detour to reach region $r3$, r_4, and r_5. For example, there were four taxis traveling from r_1 to r_5 in a 15-minute time interval before the traffic control (that occurred at 9:30). However, the volume of traffic decreased to zero after the traffic control. Similar changes happened on other paths. This example shows that our framework can detect anomalies caused by sudden increase or decrease of traffic. This example also demonstrates the ability of our framework in revealing the possible underlying cause of a phenomena. In this particular instance, the problem does not lie in the region even though the anomaly occurred there. Without the identification of the routes contributing to the anomaly it would have been very difficult to conclude that the major problem was in the traffic control areas (marked red).

2) Semi-Synthetic Experiments: To scale up the real case study mentioned above, we have also carried out semi-synthetic experiments based on real data to test the precision and recall of our method. Specifically, we still formulate a link graph (like Figure 5 right) and a corresponding link-route matrix A based on the real taxi trajectory data received in a time window. We then manually eliminate some normal links from the graph, e.g., $L5$, as illustrated in Figure 10 and distribute the traffic on $L5$ to another path, for instance, the shortest path connecting $r3$ and $r6$ (assuming $r3 \rightarrow r4 \rightarrow r6$). Accordingly, the original paths $P1$ and $P2$ will be modified to $P1'$ and $P2'$. To achieve this, we first perform the first step of the proposed anomaly detection method on the data. The detected anomalous links will not be picked out for the semi-synthetic evaluation. In the meantime, we properly select a normal link to be cut (like $L5$), making sure the traffic volume on the link is big enough to deviate the traffic on the alternative path (e.g., $r3 \rightarrow r4 \rightarrow r6$) from normal status. We first check whether the link $L3$ and $L4$ as well as the removed link $L5$ can be detected as an anomaly. Secondly, we test if our method can find the root cause contributing to the anomaly of $L3$ and $L4$, i.e., $P1'$ and $P2'$. We study the precision and recall of each step as a function of the volume of traffic (i.e., number of taxis traveling) on the link we removed. Figure 11 A) and B) respectively shows the precision and recall of our method in detecting anomalous links (i.e., the first step using PCA). The horizontal axis denotes the traffic volume on the link we cut. We randomly chose 200 time slots from

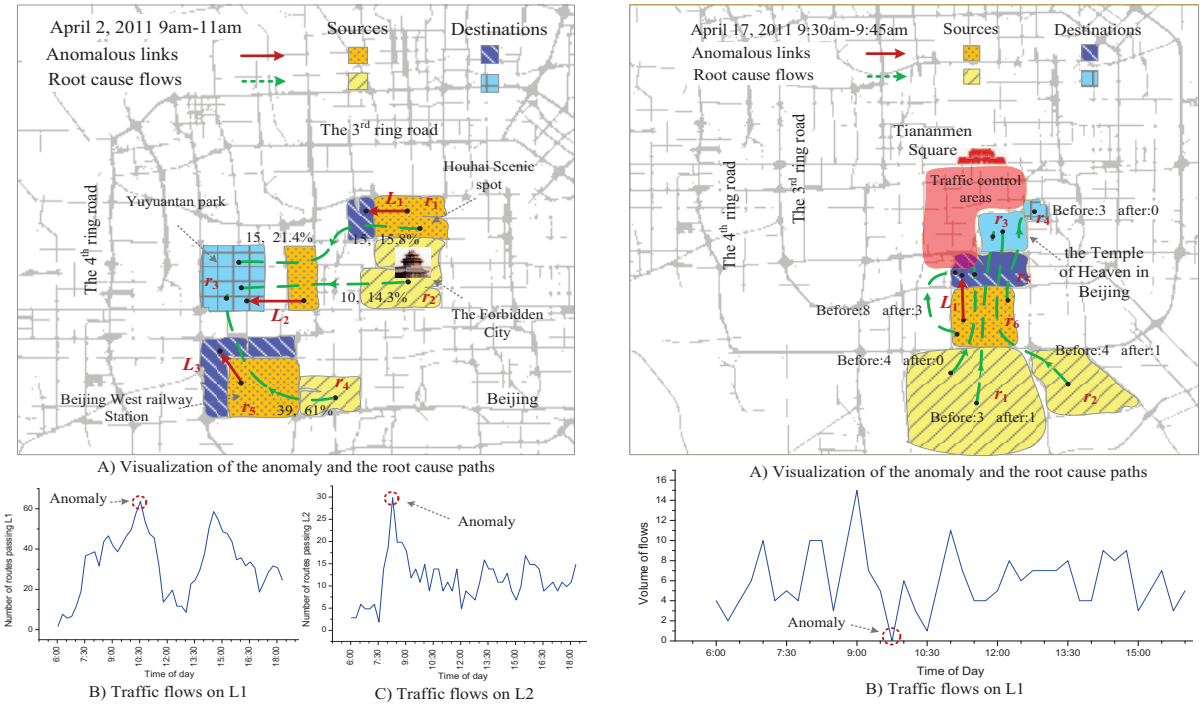


Figure 9. (I) The traffic changes due to the Sakura festival is an example of anomaly caused by an increase in traffic. (II) The Beijing marathon results in an anomaly due to a reduction in traffic.

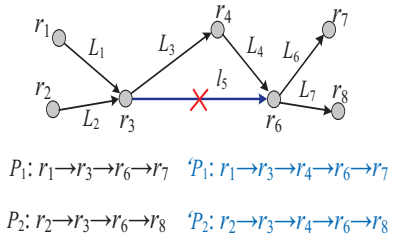


Figure 10. The semi-synthetic setup to evaluate the precision and recall of our method.

the data (including workdays and holidays) and select three links of each volume in each slot to test (as demonstrated in Figure 10). That is, we performed 3*200=600 tests for each volume of link. Later, we calculate the average precision and recall for each volume. Generally, our method becomes more capable of detecting anomalous links when the eliminated link has a larger traffic volume. Meanwhile, a relatively large window size (e.g., w=2hours) makes our method more accurate than using a smaller one (e.g., w=1hour). However, further increasing the window size (e.g., w=4hours) does not help any more. This is in line with our intuition that observing during a longer time window is more likely to identify anomalies accurately; however, a very long observing time window is not necessary and would bring noise into the inference.

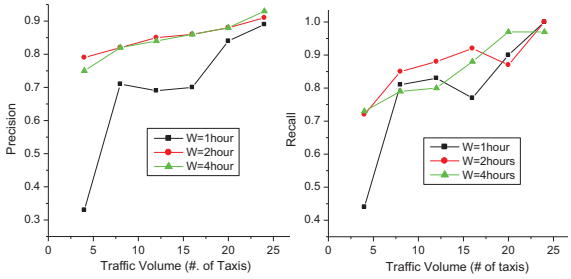


Figure 11. Effectiveness of step 1 (time bin=15min)

Figure 12 A) and B) show the effectiveness of inferring the root cause of the detected anomalous links (i.e., the second step). As a result, CVX-L1 method outperforms CVX-L2 in both precision and recall, demonstrating its advantages over the latter. Furthemore, CVX-L1 has a relatively stable performance and is is not too sensitive to the changes in traffic volume on a link.

V. Related Work

We review four strands of research which are relevant to this paper. These are (i) mathematical modeling of general traffic networks , (ii) transportation systems analysis, (iii) network anomaly detection techniques and (iv) analysis of GPS data. The first three strands have a rich history and we

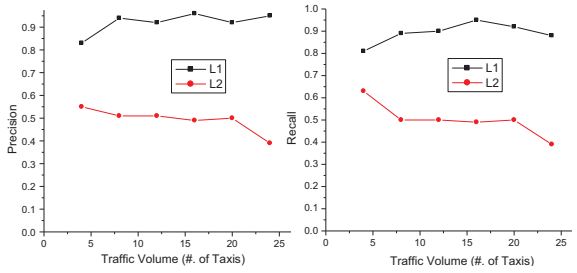


Figure 12. Effectiveness of step 2 (time bin=15min, W=2hours)

will only be able to highlight the salient features relevant to this paper.

A. Mathematical Modeling of Traffic Networks

We review the mathematical modeling aspects of traffic networks from [9]. As noted before the basic primitive is the link-route matrix A as shown in Figure 2(E), along with the link flow and route flow vectors b and x respectively. The flow on each link y_i is given by

$$b_i = \sum_j A_{ij} x_j$$

Another important primitive (in mathematical modeling) is the origin-destination pair and route matrix, H which is also an $\{0,1\}$ incidence matrix. Here , $H(s,r)$ captures the information that route r links the origin-destination pair s. In our work we do not explicitly model H as we are able to infer this information from route flows. The congestion on each link b_i is captured by the delay function $D_i(b_i)$. The delay function can be calibrated for each link but is generally an increasing function of b_i. Now the time to travel along each route j is given by the expression

$$\sum_i D_i(b_i) A_{ij}$$

Suppose there are two routes, r and r' that can be used to serve an origin destination pair s. The rational for a driver to choose the route r such that compared to every other route r'

$$\sum_i D_i(b_i) A_{ir} \leq \sum_i D_i(b_i) A_{ir'}$$

In a remarkable result due to Beckmann et. al. [1], it was shown that the local strategy of Wardrop equilibrium is the optimal solution of the global optimization problem

$$\begin{array}{rl} \text{minimize} & \sum_i \int_0^{b_i} D_i(u)du \\ \text{subject to} & Hx = s, Ax = b \\ \text{and} & x, b \geq 0 \end{array}$$

Much of the mathematical research in traffic networks (both transportation and internetworking) is concerned with the analysis of how local optimal choices can be captured by a global optimization problem. For example, a similar characterization has been obtained for the TCP protocol for data traffic. Mathematical modeling of traffic is essentially a "forward" exercise. The role of data is to calibrate the model (e.g., the delay function). In data mining, we are more interested in the "inverse" problem: how can we use data to infer information about events which are causing traffic to deviate from equilibrium.

B. Network Anomaly Detection

Our framework is closest to a body of work in the networking community. The starting point is the paper by Lakhina et. al. [11], which introduced the use PCA for detecting network anomalies like denial of service attacks, flash crowds, ping flood etc. PCA was used to exploit spatial and temporal correlation between link traffic. Anomalies were discovered by identifying time buckets which were mostly resided in the subspace spanned by the low eigenvectors (i.e., eigenvectors corresponding to low eigenvalues) of the covariance matrix of LL^T where L is the link-time matrix. These time buckets were labeled as anomalous. In our case we look for link anomalies and thus we work in the eigenspace of the matrix $L^T L$. In a subsequent paper, Zhang et. al. [17] combined network anomaly detection with optimization (including L_1 optimization) to identify source-destination which caused the anomalous time bins. However in both these and other papers in network anomaly detection, the objective is to identify network anomalies and also their potential origin-destination pairs. In our case, we begin with almost no information about the events or even the type of events that are causing traffic perturbations.

C. Transportation and Traffic Analysis

In the transportation systems literature, the problem of relating link traffic to source-destination pairs is called the *observability problem.* The standard text in this area is *Transportation Systems* by Cascetta [5]. In this community the source of data is still primarily sensors which are embedded in roads and measure volume and occupancy rates of each link. To the best of our knowledge the use of L_1 optimization for inference of sparse route vectors has not been used in the community.

There is some recent work [7], [13] that detects traffic jams on road surfaces using GPS traces of vehicles. Our framework is different from these techniques in two parts. First, the traffic anomalies we detect are far beyond traffic congestions, e.g., it could be a sudden decrease caused by a traffic control. Sometime, an anomaly occurs even if a road is not congested. Second, we study the traffic between regions instead of on road surfaces. By this means, we can not only reduce the complexity of modeling city-wide traffic but also are to detect the root-cause of traffic anomalies.

Finally we would like to report some recent work in this area which uses a similar data set. Zheng et. al. [18] have used the data set to investigate the connectivity flaws in the road network. Liu et. al. [12] have used frequent subgraph mining to discover anomalous links for each time interval and then connect the anomalies across time intervals to form outlier trees. In this paper we also look for anomalies but the key difference is the use of L_1 machinery to elicit the cause of anomalies discovered. Similary [14] et. al. have proposed the use of likelihood ratio tests to determine regions where the traffic volume has deviated substantially from the norm. Again, this work is algorithmic and does to attempt to explain the cause of anomalies. Finally we would like to note that our work contributes towards the growing body of literature on Urban Computing [19].

VI. Summary and Conclusion

In this paper we have proposed a framework to analyze a large GPS data set obtained from over thirty thousand taxis in Beijing over a three month period. Our framework has two steps: mining and optimization. In the mining step we have used Principal Component Analysis (PCA) to discover link anomalies from GPS data. From the link anomalies it is difficult to infer about what *caused* the anomalies to occur. In order to gain further insights we used the link-route incidence matrix to formulate an L_1 optimization problem. The sparse solution of the optimization problem gives a candidate set of routes which can be used to explain why anomalies occur. We give several real examples of such anomalies.

References

[1] M. Beckmann, C. McGuire, and C. Winsten. Studies in the economics of transportation. In *Cowles Commission Monograph*. Yale University Press, 1956.

[2] D. Brauckhoff, K. Salamatian, and M. May. Applying pca for traffic anomaly detection: Problems and solutions. In *IEEE INFOCOMM*, 2009.

[3] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[4] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[5] E. Cascetta. *Transportation Systems Analysis: Models and Applications*. Springer optimization and its applications. Springer, 2009.

[6] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43:129–159, January 2001.

[7] W. Dong and A. Pentland. A network analysis of road traffic with vehicle tracking data. In *AAAI Spring Symposium*, 2009.

[8] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

[9] F. Kelly. The mathematics of traffic. In T. Gowers, editor, *The Princeton Companion to Mathematics*, pages 862–860. Princeton University Press, 2008.

[10] F. Korn, H. V. Jagadish, and C. Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. In *SIGMOD Conference*, pages 289–300, 1997.

[11] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *Proceedings of the ACM SIGCOMM*, pages 219–230, 2004.

[12] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing. Discovering spatio-temporal causal interactions in traffic data streams. In *KDD*, pages 1010–1018, 2011.

[13] R. Ong, F. Pinelli, R. Trasarti, M. Nanni, C. Renso, S. Rinzivillo, and F. Giannotti. Traffic jams detection using flock mining. In *ECML/PKDD (3)*, pages 650–653, 2011.

[14] L. X. Pang, S. Chawla, W. Liu, and Y. Zheng. On mining anomalous patterns in road traffic streams. In *ADMA (2)*, pages 237–251, 2011.

[15] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[16] J. Yuan, Y. Zheng, and X. Xing. Discovering regions of different functions in a city using human mobiity and pois. In *KDD*, 2012.

[17] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network anomography. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 30–42, 2005.

[18] Y. Zheng, Y. Liu, J. Yuan, and X. Xie. Urban computing with taxicabs. In *Ubicomp*, pages 89–98, 2011.

[19] Y. Zheng and X. Zhou, editors. *Computing with Spatial Trajectories*. Springer, 2011.