Temporal Outlier Detection in Vehicle Traffic Data

Xiaolei Li [1*], Zhenhui Li [2], Jiawei Han [2], Jae-Gil Lee [3*]

[1]*Microsoft Corporation, Seattle, WA, United States*
[2]*Department of Computer Science, University of Illinois, Urbana, IL, United States*
[3]*IBM Corporation, San Jose, CA, United States*
[1]xiaoleil@microsoft.com
[*]Work done while at University of Illinois

***Abstract*— Outlier detection in vehicle traffic data is a practical problem that has gained traction lately due to an increasing capability to track moving vehicles in city roads. In contrast to other applications, this particular domain includes a very dynamic dimension: *time*. Many existing algorithms have studied the problem of outlier detection at a single instant in time. This study proposes a method for detecting *temporal outliers* with an emphasis on historical similarity trends between data points. Outliers are calculated from drastic changes in the trends. Experiments with real world traffic data show that this approach is effective and efficient.**

I. Introduction

With the increasing popularity of GPS and RFID devices, the tracking of motor vehicles and moving objects in general has become a reality in many cities. One very practical and important problem in vehicle traffic analysis is outlier detection. A typical definition of an outlier is "*an observation (or a set of observations) which appears to be inconsistent with the remainder of that set of data* [2]." This rather vague definition can lead to many different outlier detection algorithms. This work will primarily use the application scenario of *detecting temporal outliers in vehicle traffic data*. More specifically, it seeks to detect outlier behavior in the set of road segments of the traffic data and not individual moving objects. Consider *only* the solid line X in Figure 1, which shows a toy example of load (count) of vehicles on road segment X over the course of several days. From this line alone, there does not seem to be any abnormal behavior. But suppose additional information about other road segments in the city are given. In light dashed lines are many other road segments that have similar loads as road segment X from July 1st to July 4th. On July 4th, however, they all show an increase in load while X remains the same. The question of outliers has now become much trickier. X has now just become an outlier. Recall the outlier definition regarding "inconsistency." This example and the rest of this work use historically similar neighbors as the basis for consistency comparisons.

This paper presents the **Temporal Outlier Discovery** or TOD framework for detecting *temporal outliers*. In contrast to other algorithms, this method utilizes agglomerated temporal information of the entire dataset to detect outliers. At each time step, each road segment checks its similarity vs. other road segments, and the historical similarity values are recorded in a *temporal neighborhood vector* at each road segment. Outliers are calculated from drastic changes in these vectors.

This work was supported in part by the U.S. National Science Foundation NSF IIS-08-42769 and BDI-05-15813, and by the Boeing company. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

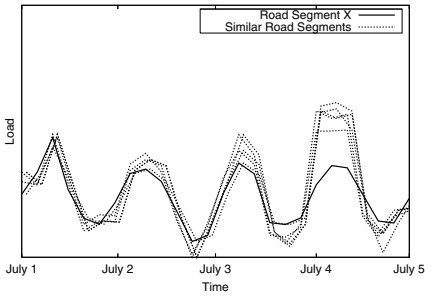

Fig. 1. Historical Load on Many Road Segments

II. Road Segment Representation

Let the road network be represented with a directed graph $G = (V, E)$ where the set of vertices V represent street intersections and the set of edges E represent road segments. Then, the most basic input data is a set of timestamp and edge tuples: $\{(t, i), \ldots\}$ where t is some time value and i is some edge in E. Every edge in E is mapped to a point in some feature space. Let the set of used features be $\mathcal{F}$, then every road segment is mapped to a point in $\mathcal{R}^{|\mathcal{F}|}$. Table I shows an example of four road segments with two features values indicating the average speed on them during the morning and afternoon hours of a day. Because the data changes temporally and temporal outlier is the topic at hand, these two feature values are updated every day (or any other time period). Table I shows 3 consecutive days of feature values.

III. Temporal Outlier Detection

Outlier analysis in TOD depends heavily on the time dimension. A prerequisite for this is the existence of some temporal patterns in the data. Figure 2 shows the average percent deviation in average daily speed and average daily load (*i.e.*, count) of taxicabs on over 20,000 road segments in the San Francisco area during the month of July 2006; The

	Day 1		Day 2		Day 3	
	AM Speed	PM Speed	AM	PM	AM	PM
1	3	20	3	19	10	20
2	5	21	3	21	4	22
3	2	20	3	23	3	21
4	15	32	4	23	20	20

TABLE I
SAMPLE FEATURE SPACE

graph shows the average percent change from the previous day for all road segments. There are over 33 million recorded vehicle road segment movements in the data set.

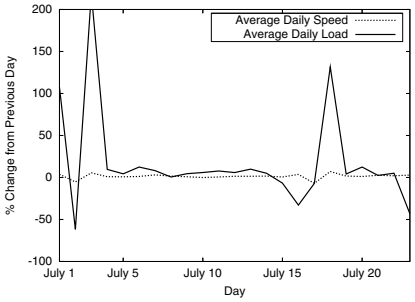

Fig. 2. Stability of Average Daily Speed and Load

As the figure shows, the average daily speed is very stable (modulo the spikes, which are caused by data collection errors) day-to-day. Though not particularly surprising, this is good news for temporal outlier detection. It shows that there are stable trends and they can be used as a basis for outlier detection. The TOD framework uses the following method of outlier detection. Instead of evaluating the trends of the measure(s) directly and *singularly*, trends of similar neighbors based on the measure(s) are evaluated. Figure 3 shows a confirmation of this observation from real world data. In it, neighbor road segments according to speed ($\pm$ 5MPH) and load ($\pm$ 50 cars) of each road segment is calculated on every day. Then, the average percent change in the size of a continual intersection of this neighbor set of all road segments is plotted against time. The neighborhoods are very stable with an average deviation of less than 5%. More interestingly, around July 16th and July 17th where data loss caused huge spikes in Figure 2, Figure 3 barely shows any effect. This is because the data loss was global and entire neighborhoods shifted together. As a result, locality of road segments in the feature space is preserved.

The outlier detection scheme in TOD is motivated by the stable neighborhoods. Given a road segment with a historically stable set of neighbors (in feature space, not physical space), an outlier is loosely defined as a drastic change in the membership of this set. The power of this method vs. a method that measures only the singular road segment is that it is robust to population shifts.

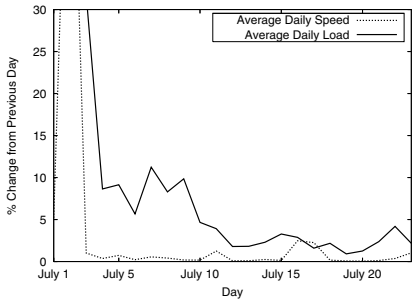

Fig. 3. Speed and Load Neighborhood Stability

A. Temporal Neighborhood Vector

***Definition** 1 (Temporal Neighborhood Vector):* Every edge i maintains a vector $\vec{v}_i$ to record historical similarities to other edges. The length of this vector is $N = |E|$. Every j^{th} value, $v_{i,j} \in \mathbb{R}$, records the historical similarity between edge i and edge j. A large $v_{i,j}$ indicates high historical similarity and vice-versa. Initially, all vector values are set to 0.

At every time step, each value $v_{i,j}$ is adjusted up or down depending on the similarity between edge i and j at that instant. TOD uses a threshold to determine similarity.

***Definition** 2 (Instantaneous Similarity):* Let $d(i, j)$ be a distance function defined on the feature space. Then, edges i and j are similar with respect to an instant in time and the particular feature values at that instant (*i.e.*, instantaneously similar) if $d(i, j) \leq \theta$.

The temporal neighborhood vector values are updated differently based on whether instantaneous similarity holds true or false. As a simple example, consider the Day 1 data from Table I. Let $d(i, j)$ be the L-∞ distance (*i.e.*, maximum absolute difference between feature values). Table II shows the L-∞ distance between pairwise edges on Day 1. Let $\theta = 5$. In this case, edges 1, 2, and 3 are instantaneously similar to each other while edge 4 is not to any other edge. Assume, for now, that each $v_{i,j}$ gets incremented by 1 if edges i and j are similar and decremented by 1 otherwise (with a minimum of 0). The temporal neighborhood vector of edge 1 is then $\langle 1, 1, 1, 0 \rangle$.

Edge Pairs	L-∞ Distance
L-∞(1, 2)	2
L-∞(1, 3)	1
L-∞(1, 4)	12
L-∞(2, 3)	3
L-∞(2, 4)	11
L-∞(3, 4)	13

TABLE II
EDGE SIMILARITY VALUES FOR DAY 1

B. Temporal Vector Update Rules

With instantaneous similarity defined, the next step is then updating the neighborhood vectors temporally, where time is essentially a sequence of instants. As mentioned briefly before, the amount of change in $\vec{v}_i$ at each time step is the measure

of anomaly or "outlier-ness" for edge i. Thus, it is critically important to have proper update rules. TOD uses the following intuition based on the existing similarity values.

1) If two edges are historically similar, then a new instantaneous similarity can be noted lightly.
2) If two edges are historically similar, then a new instantaneous dissimilarity should be noted heavily.
3) If two edges are not historically similar, then a new instantaneous similarity should be noted heavily.
4) If two edges are not historically similar, then a new instantaneous dissimilarity can be noted lightly.

It turns out that exponential functions conveniently capture the above intuition. Let the period of update be daily and consider the similarity value $v_{i,j}^{d-1}$ between edges i and j on day $d-1$. If these two edges are instantaneously similar on the next day d, the reward is defined as

$$reward(i,j,d) = \alpha_1^{v_{i,j}^{d-1} - \alpha_2} \qquad \alpha_1 < 1.0,\ \alpha_2 \geq 0 \quad (1)$$

For progressively larger $v_{i,j}^{d-1}$ values, the reward naturally becomes smaller. The update function for $v_{i,j}^{d}$ is then $v_{i,j}^{d} = v_{i,j}^{d-1} + reward(i,j,d)$. The same equation applies to penalties as well. If edges i and j are instantaneously dissimilar on day d, the penalty is defined as

$$penalty(i,j,d) = \beta^{v_{i,j}^{d-1}} \qquad \beta > 1.0 \quad (2)$$

For progressively larger $v_{i,j}^{d-1}$ values, the penalty becomes larger exponentially. The update function for $v_{i,j}^{d}$ is then $v_{i,j}^{d} = v_{i,j}^{d-1} - penalty(i,j,d)$. Continuing the running example, the updated temporal vectors from Day 1 until Day 3 from Table I are shown in Table III. α_1 is set to 0.9, α_2 is set to 0, β is set to 1.1, θ is set to 5, and $d(i,j)$ is set to L-∞. There is a minimum of 0 to any $v_{i,j}$ value.

Edge	Temporal Neighborhood Vectors		
	Day 1	Day 2	Day 3
1	$\langle 1,1,1,0\rangle$	$\langle 1.9,1.9,1.9,1\rangle$	$\langle 2.7,0.7,0.7,0\rangle$
2	$\langle 1,1,1,0\rangle$	$\langle 1.9,1.9,1.9,1\rangle$	$\langle 0.7,2.7,2.7,0\rangle$
3	$\langle 1,1,1,0\rangle$	$\langle 1.9,1.9,1.9,1\rangle$	$\langle 0.7,2.7,2.7,0\rangle$
4	$\langle 0,0,0,1\rangle$	$\langle 1,1,1,1.9\rangle$	$\langle 0,0,0,2.7\rangle$

TABLE III
DAY 1–3 TEMPORAL NEIGHBORHOOD VECTORS

C. Temporal Outlier Scoring

Outliers in TOD are measured from *drastic* changes in the temporal neighborhood vectors. Intuitively, the most drastic or abnormal changes are ones that differ the most from historical values. Furthermore, the more stable the historical values are previously, the more the change should contribute to the overall outlier score. Fortunately, this intuition is easily captured in the reward and penalty equations in the previous section. Because $v_{i,j}^{d-1}$ is in the exponent of Equations (1) and (2), the big rewards and penalties will come from previously stable trends (either similar or dissimilar). The outlier score of edge i on a particular day is then equal to the sum of rewards and penalties. Let $v_{i,j}^{d}$ represent the value of $v_{i,j}$ on day d. Then, the *outlier score* of edge i on day d, $OS(i,d)$, is defined as

$$OS(i,d) = \sum_{j=1, j\neq i}^{N} \left| v_{i,j}^{d} - v_{i,j}^{d-1} \right| \quad (3)$$

For a given edge, Equation (3) is calculated with respect to all other edges in the road network. This may go against the natural intuition in spatial problems to use spatial proximity as one of the criteria for choosing comparative edges. This approach seems natural but contains a serious flaw: it could potentially hide anomalies that affect a region of edges. For example, suppose there is a very serious traffic accident that affects many edges in a local area. A local approach might miss it altogether. Continuing the previous examples, consider the outlier scores of Table III. $OS(1,2) = |1.9-1| + |1.9-1| + |1.0-0| = 2.8$ and so forth.

IV. EXPERIMENTS

Real world moving object data was used to test the effectiveness of TOD. 24 days of moving taxicab data in the San Francisco area were collected during the month of July in 2006[1]. In all, there were over 800,000 separate trips, 33 million road segment traversals, and 100,000 distinct road segments. Figure 4 shows an outlier with respect to load (*i.e.*, count) of vehicles. The neighboring segments have a stable trend (427 neighbors with daily standard deviation under 1.0) from July 3rd to July 13th while the outlier road segment's load increases significantly on July 9th and 13th.

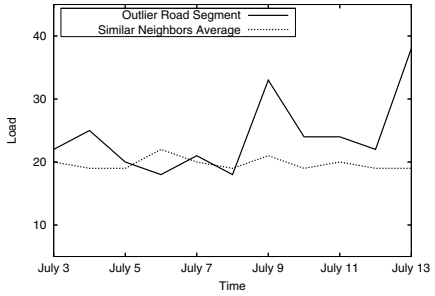

Fig. 4. Load Outlier

Figure 5 shows another outlier with respect to load. In this case, the cause is opposite than that of Figure 4. The load of the neighbors (107 of them) increases on July 6th and 7th (weekend) while the road segment in question remains stable. This is rather strange because the road segment is very similar to its neighbors on weekdays. And yet, it did not receive the weekend bump in load due to some unknown reason.

The basis for consistency comparisons in TOD are historical similarities between road segments. In contrast to using direct historical trends in the measure itself, this is more powerful since sometimes trends may not exist in the measure but do in the similarities. Figure 6 shows a non-outlier in TOD.

[1]Unfortunately, this dataset is not available publicly.

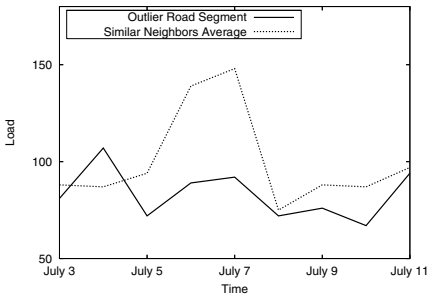

Fig. 5. Load Outlier

On July 7th and 8th, the load increases significantly when compared to previous data. A naïve method is likely to report these outliers. But when compared to its historically similar neighbors, the increase is global and rather normal. Those two days are actually weekend days (Friday and Saturday) and the universal increase in load is reasonable.

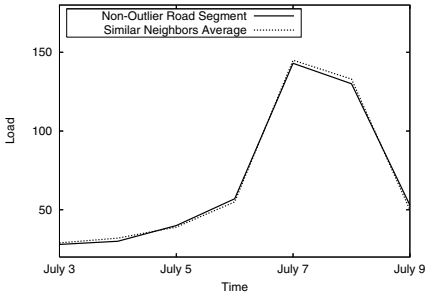

Fig. 6. Non-Outlier

Finally, the efficiency of TOD is tested. Figure 7 shows the running times of TOD as a function of the number of days processed. There was a total of 33 million road segment traversals spread throughout 24 days, which makes an average of roughly 1.4 million road segment traversals per day. As the figure shows, processing these traversals and updating the temporal neighborhood vectors is a linear time operation with respect to the number of days.

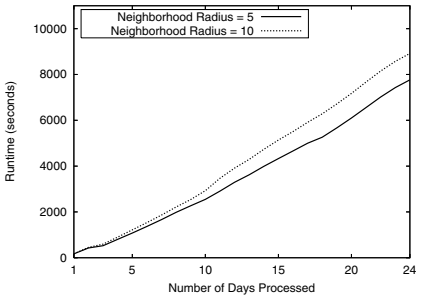

Fig. 7. Efficiency vs. Number of Days

V. Related Work

Much focus has been put on moving object databases lately, specifically in the context of road networks or constrained networks. Some have also addressed the outlier detection problem [8], [7]. In comparison to TOD, they seek outlier *objects* instead of road segments. That is, a single object will be labeled as an outlier at a specific location due to its abnormal trajectory. Clustering moving objects [4] and time series [6] are also related to this work. However, there is a big difference in that TOD's "clustering" is updated everyday while time series clustering/classification is usually applied to the entire dataset.

Outlier detection algorithms in other contexts are related as well. In traditional data points without a temporal dimension, many algorithms [3], [9] have been proposed. TOD has similarities to one [3], which detects *local* outliers. The temporal neighborhood vector can be viewed as a historical record of local neighbors. In time series research, algorithms have been proposed to find outliers [5], [11]. One algorithm [5] defines outliers as time series that have very distant nearest neighbors. Compared to TOD, this definition is very simplistic in the sense that it only measures entire time series and do not consider temporal stability within the time series. Streaming data is another domain where outliers have been studied. One recent study [10] addresses the problem of outliers in sensor data. The definition of an outlier is very different though. TOD focuses on temporal outliers while [10] focuses on traditional outliers within an instant of time and how to efficiently compute them repeatedly across many instants. And finally, outliers have been studied in the context of text streams. BlogScope [1] is similar to TOD in that it also detects "stable clusters." But the points in the clusters are keywords and connections between them are made based on document co-occurrences. But more importantly, the temporal clustering is hard and outliers are found through an expensive graph search.

References

[1] N. Bansal and N. Koudas. Blogscope: A system for online analysis of high volume text streams. In *VLDB'07*.
[2] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.
[3] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *SIGMOD'00*.
[4] P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *SSTD'05*.
[5] E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *ICDM'05*.
[6] E. J. Keogh and M. J. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *KDD'98*.
[7] J.-G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *Proc. 2008 Int. Conf. Data Mining (ICDE'08)*, Cancun, Mexico, April 2008.
[8] X. Li, J. Han, S. Kim, and H. Gonzalez. Roam: Rule- and motif-based anomaly detection in massive moving object data sets. In *SDM'07*.
[9] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *ICDE'03*.
[10] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. In *VLDB'06*.
[11] D. Yankov, E. Keogh, and U. Rebbapragada. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. In *ICDM'07*.