10-2018

Traffic-Cascade: Mining and visualizing lifecycles of traffic congestion events using public bus trajectories

Agus Trisnajaya KWEE
Singapore Management University, aguskwee@smu.edu.sg

Meng-Fen CHIANG
Singapore Management University, mfchiang@smu.edu.sg

Philips Kokoh PRASETYO
Singapore Management University, pprasetyo@smu.edu.sg

Ee-peng LIM
Singapore Management University, eplim@smu.edu.sg

Citation

KWEE, Agus Trisnajaya; CHIANG, Meng-Fen; PRASETYO, Philips Kokoh; and LIM, Ee-peng. Traffic-Cascade: Mining and visualizing lifecycles of traffic congestion events using public bus trajectories. (2018). *CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge Management: Torino, Italy, October 22-26*. 1955-1958.
Available at: https://ink.library.smu.edu.sg/sis_research/4260

Traffic-Cascade: Mining and Visualizing Lifecycles of Traffic Congestion Events Using Public Bus Trajectories

Agus Trisnajaya Kwee
Living Analytics Research Centre
Singapore Management University
aguskwee@smu.edu.sg

Meng-Fen Chiang
Living Analytics Research Centre
Singapore Management University
mfchiang@smu.edu.sg

Philips Kokoh Prasetyo
Living Analytics Research Centre
Singapore Management University
pprasetyo@smu.edu.sg

Ee-Peng Lim
Living Analytics Research Centre
Singapore Management University
eplim@smu.edu.sg

ABSTRACT

As road transportation supports both economic and social activities in developed cities, it is important to maintain smooth traffic on all highways and local roads. Whenever possible, traffic congestions should be detected early and resolved quickly. While existing traffic monitoring dashboard systems have been put in place in many cities, these systems require high-cost vehicle speed monitoring instruments and detect traffic congestion as independent events. There is a lack of low-cost dashboards to inspect and analyze the lifecycle of traffic congestion which is critical in assessing the overall impact of congestion, determining the possible the source(s) of congestion and its evolution. In the absence of publicly available sophisticated road sensor data which measures on-road vehicle speed, we make use of publicly available vehicle trajectory data to detect the lifecycle of traffic congestion, also known as **congestion cascade**. We have developed **Traffic-Cascade**, a dashboard system to identify traffic congestion events, compile them into congestion cascades, and visualize them on a web dashboard. Traffic-Cascade unveils spatio-temporal insights of the congestion cascades.

CCS CONCEPTS

• **Information systems** → *Data mining*; • **Computing methodologies** → *Anomaly detection*; • **Human-centered computing** → *Visual analytics*; *Geographic visualization*;

KEYWORDS

traffic congestion; data mining; anomaly detection; visualization

ACM Reference Format:
Agus Trisnajaya Kwee, Meng-Fen Chiang, Philips Kokoh Prasetyo, and Ee-Peng Lim. 2018. Traffic-Cascade: Mining and Visualizing Lifecycles of Traffic Congestion Events Using Public Bus Trajectories. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269216

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269216

1 INTRODUCTION

Road transportation has always been a critical part of city development. It connects people from different locations of a city and supports many types of urban activities including economic and social activities. While more and more developed cities start utilizing subway systems, road transportation remains to be the main mode of land transportation. City residents therefore still pay significant amount of attention to the health of road transportation.

One major concern of using road transportation in developed cities is traffic congestion events. Statistics have shown that many man-hours have been wasted in traffic congestion. It not only reduces productivity, but also contributes to environmental problems. There is certainly an important need for automated traffic management system to support three essential functions, namely, (a) detection, (b) analysis, and (c) lifecycle exploration of road traffic congestion.

Many traffic management systems come with dashboard interfaces that tap on specially instrumented road sensor data or privately gathered vehicle trajectory data [11]. Recent work also utilizes telco data for traffic analytics [4]. These data are generally not available to the general public as they are incorporated into custom-built dashboard systems for use in private companies or the transportation departments of some government agencies. A few public web-based systems have been developed to show online live traffic on road. Waze [1] and Google Map [2] show live road traffic status based on the speed of vehicles on roads. These systems rely on crowd-sourced speed data from private vehicles on the roads. Recent studies utilize tweet content as traffic signal to detect traffic events [5] and traffic congestions [2, 9, 10]. A visualization of traffic event detection built by [6] also utilizes Twitter data. However, crowd-sourced speed data and social media signals may not available all the time on all road segments. On one hand, the method to collect crowd-sourced speed data may not be easy to replicate in many cities. On the other hand, such systems support primarily the detection functions, but not analysis and lifecycle exploration of traffic congestions. In this paper, we focus on a web dashboard

[1] https://www.waze.com/livemap
[2] https://maps.google.com/

system that supports multiple functions to visualize lifecycles of traffic congestions.

Contributions. Unlike the previous congestion detection works, we model congestion events as part of larger congestion cascades. Each **congestion cascade** captures the lifecycle of congestion growing and shrinking processes along connected road segments. With congestion cascades modeling both temporal and spatial semantics of a set of closely related congestion events, users now have much fewer congestion cascades than transient congestion events to examine in a web-based dashboard. On the other hand, users could leverage on the larger UI visualization bandwidth to examine the detected congestions when analyzing road traffic health. The state-of-the-art systems [6, 11] does not support the construction of congestion cascade and thus cannot capture the formation and lifecycle of a closely related congestion events. Recent work [1] on tracking the evolution of traffic congestion focuses on monitoring congested partitions only, and it does not have visualization system. Traffic-Cascade, a proposed web-based dashboard system, aims to address the above limitations.

Powered by a recently developed *Bus Trajectory based Congestion Identification* (BTCI) framework [3], Traffic-Cascade is able to detect congestion cascades using public bus trajectory data, and to visualize the detected cascades. As an online system, Traffic-Cascade makes congestion cascade detection and analysis available to all public users. Traffic-Cascade shows that public bus data trajectory data represents a cheap but viable input data for detecting congestion cascades, and that congestion cascade is a more appropriate granularity to analyze the set of congestion events.

Paper Outline. In the following section, we discuss the system overview of Traffic-Cascade in Section 2. We introduces our trajectory data collection in Section 3. Details of Traffic-Cascade components are presented in Sections 4 and 5. We describe implementation and various scenarios supported for the demo in Section 6. Finally, we conclude our work in section 7.

2 SYSTEM OVERVIEW

Figure 1 shows high-level overview of Traffic-Cascade system. The core component of the system is BTCI framework [3]. The system requires bus trajectory data which we collect from public web API service. The BTCI framework analyzes and mines the trajectory data in two stages to identify traffic congestions. First, BTCI extracts congested segments based on speed information derived from bus trajectory data, and then clusters the extracted segments to identify traffic congestion events. Each cluster represents a traffic congestion cascade for each congestion event. Lastly, a web module visualizes the identified events from BTCI to the users. We discuss each part in more details in the next three sections.

3 TRAJECTORY DATA COLLECTION

Traffic-Cascade utilizes GPS location traces of thousands public buses to identify traffic congestion events. From sequences of bus location snapshots, we construct bus trajectory data. Our trajectory data are constructed using public bus location information from public web service API provided by Singapore Land Transport Authority (LTA) [3]. This public API consumes bus stop ID, and returns arrival time and current bus location of each bus service approaching that particular bus stop. Utilizing this public web API and a list of all bus stops, we are able to construct a bus trajectory dataset covering a period of time interval.

[3]https://www.mytransport.sg/content/mytransport/home/dataMall.html

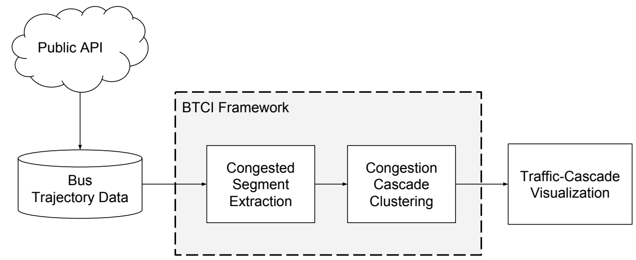

Figure 1: Traffic-Cascade System Overview

For this demo, we utilize bus trajectory data from all buses passing through Singapore expressway from two periods of time: (1) *May-June* trajectory data from 20 May 2016 to 20 June 2016, (2) *July* trajectory data from 1 July 2016 to 31 July 2016. The data are generated by aggregating queries from 18 bus stops in Singapore expressway. In total, we collected about 11.8 million bus locations. We then preprocess these trajectory data by dividing them into spatio-temporal segments.

- **Spatial Aspect.** We derive 82 paths (i.e., sequences of road segments where the traffic flows) where each path covers four bus stops.
- **Temporal Aspect.** We divide the data into 15-minute sliding time windows with 1-minute shift.

A spatio-temporal segment consists of information about a path in 15-minute time window. In total, we have 118,080 segments (i.e., 82 paths $\times$ 60 minutes $\times$ 24 hours) each day. After removing segments with low speed observations (i.e., < 10 observations), we obtain 87,052 segments.

4 BTCI FRAMEWORK

We formally define congestion cascade problem addressed by BTCI as follows. Given a trajectory data D collected over a period of time T and historical trajectory data H, BTCI identifies a set of congestion cascades $\{CG_1, CG_2, ..., CG_n\}$ within time T. Each congestion cascade is defined as a cluster of spatio-temporal congested segments which characterize the formation and lifecycle of the congestion event. Each congestion cascade consists of connected congested road segments sharing both spatio-temporal closeness and coherent traffic flow direction criteria (see definition 4.1). Our unit of analysis is a segment as defined in Section 3.

Definition 4.1. A **congestion cascade** is a weighted graph of congested segments: $CG = \{CS_1, CS_2, ..., CS_m\}$, where a node CS_i represents a congested segment, and an edge (CS_i, CS_j) indicates that CS_j is influenced by CS_i. The influence is affected by coherent traffic flow direction and spatio-temporal closeness between CS_i and CS_j.

The first stage of BTCI, congestion segment extraction, discovers all congested segments CS. The second stage of BTCI, congestion cascade clustering, compiles all congested segments CS into a set of congested cascades $\{CG_1, CG_2, ..., CG_n\}$.

4.1 Congested Segment Extraction

A congested segment is defined as a segment with anomalous slow-speed deviation from usual speed pattern. Our idea to extract congested segments is to firstly model speed pattern for each segment to establish the statistical norm of traffic speed based on historical data H. Secondly, BTCI assigns congestion score to each segment in trajectory data D by measuring the slow-speed anomaly of a segment based on the deviation of observed speeds on target day from its norm with the same (road path, time window) pair. We use *May-June* dataset to establish the usual speed patterns for all segments, and use it to detect congestion segments in *July* dataset.

BTCI utilizes Kernel Density Estimation (KDE) [7] to derive speed model for each segment. KDE, a non-parametric approach, does not rely on a prescribed probability distribution, but uses the sum of kernel functions centered at data points to estimate the probability density. After a speed model is established for a segment S from the historical observations, the congestion score is derived as the probability of a new observation belonging to slow-speed anomaly according to the speed model on S. We use the lower tail distribution (i.e., the slowest 10% of the speed distribution) as the area of anomalous slow speeds.

The congestion score c_score of a segment S with n speed observation is then defined as, $c\_score(S) = \frac{k}{n}$, the proportion of k speed observations falling under anomalous slow speed over all speed observations in the segment. Congested segment extraction therefore identifies a set of congested segments: $CS = \{S | c\_score(S) \geq \gamma\}$, where γ denotes the user specified minimum congested segment score threshold. γ controls the degree for segments to be considered as congested. Higher γ results in less qualified c-segments. Our choice of γ is empirically determined as 0.3.

4.2 Congestion Cascade Clustering

Given a set of congested segments extracted from the previous stage, congestion cascade clustering finds a set of congestion cascades $\{CG_1, CG_2, ..., CG_n\}$ following definition 4.1.

BTCI employs a probabilistic generative model, GenClus [8], originally designed for attributed heterogeneous network clustering. GenClus enables BTCI to cluster congested segments that exhibit similar attribute values as well as strong spatial and temporal connectivity, i.e., similar congestion scores and spatio-temporal closeness. The algorithm solves a cluster optimization objective function using EM algorithm. Please refer to [3] for details.

Major clusters that are highly coherent in attribute values and spatio-temporally concentrated are promising candidates of congestion cascades. However, some congested segments in reality may be disengaged in some congestion cascades, i.e., some congested segments may not be part of any congestion cascades. These cases represent noises and introduce incoherences in the resultant clustering. We introduces two post processings to distill truly interesting congestion cascades:

Weak Congested Segment Filtering aims to filter congested segments with very small likelihood assignments to their belonging clusters. A membership threshold is introduced to eliminate such weak congested segments.

Incoherent Congested Segment Filtering aims to filter congested segments that contribute either incoherent attribute values or weak spatio-temporal adjacency to their belonging clusters. A coherence threshold is introduced to eliminate such incoherent congested segments.

5 VISUALIZATION DASHBOARD

Congestion cascades mined by BTCI capture the lifecycle of congestion events, from growing to shrinking, both spatially and temporally. A congestion cascade consists of multiple spatio-temporal segments containing speed information of road paths over time from the beginning until the end of the congestion cascade. Furthermore, user may also want to inspect the severity of a congestion event by comparing it against the usual pattern on historical data. Since the congestion cascades involve many facets, it is difficult to explore them without any tools. Therefore, we build a visualization dashboard to meaningfully display the cascades so that users are able to explore, inspect, and investigate the congestion cascades.

The dashboard takes the congestion cascades from BTCI, and it consists of three main components: (1) A list of *identified congestion cascades* at left panel, (2) Charts summarizing *temporal aspect of the selected congestion cascade* at the right and bottom panels, (3) A map visualizing *spatial aspect of the selected congestion cascade* at the middle panel, segment by segment over time. Figure 2 shows the screenshot of the dashboard.

The left panel shows a summary of each identified traffic congestion cascade: start and end points of the congested road segments, time span of the cascade, average and standard deviation of speed in the cascade, and bus services involved in the cascade. Once selected, a congestion cascade will have its detailed spatio-temporal information presented in the other panels.

Charts at right and bottom panels visualize the temporal aspect of bus speed and congestion score in a selected congestion cascade. Top-right panel shows speed distribution of selected cascade against historical norm, indicating the severity of a congestion cascade compared to the usual pattern. Bottom-right and bottom panels show bus speed and congestion score over time. Clicking the charts allows the user to jump to a particular time point and explore the spatial situation at that time point on the map.

Beside visualizing the affected road segments, the map also visualizes nearby reported road incidents to provide richer contextual information about the impact of nearby incidents to the congestion cascade. User can click each road segment on the map to get more detailed summary information about the corresponding road segment that include the affected bus services, and average and standard deviation of speed in the road segment. By clicking an incident marker on the map, the user will open an *infowindow* showing the official authority announcement on the incident. A playback button below the map enables users to view, explore, and inspect the whole formation and lifecycle of a selected congestion event.

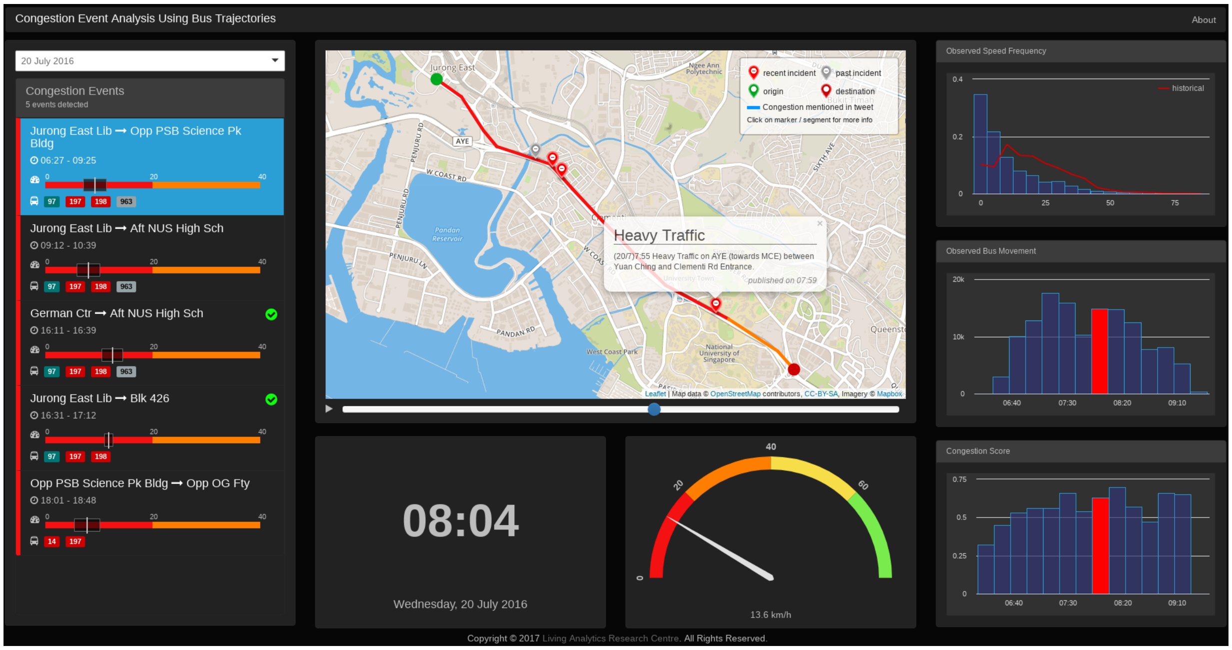

Figure 2: Screenshot of Traffic-Cascade Dashboard

6 DEMO

Implementation. We store our trajectory data in Elasticsearch [4] cluster. The core component of Traffic-Cascade, BTCI framework, is implemented in Java. All results are computed in Centos 6.8 OS VM equipped with 7 cores CPU, 70GB RAM. The visualization dashboard is developed as a node.js [5] application. The Traffic-Cascade dashboard system is available publicly at http://research.larc.smu.edu.sg/traffic-cascade/.

Scenario. The audience will be able to explore lifecycle of detected congestions in the system. More specifically, they will be able to learn the relation of temporal and spatial aspect of congestion cascades through animated playback, as well as the detailed contextual information from *infowindow* and objects in the interactive map. We hope that the demonstration can spark exciting and meaningful discussion to make traffic information system more available to general public.

7 CONCLUSION

Analyzing and exploring traffic congestion events and their impacts do not consist of only detecting snapshots of congestion events, but also capturing the lifecycle of congestion events, including growing and shrinking along road segments. Formulating the problem as congestion cascades identification enables us to capture the formation and lifecycle of congestion events. We have presented Traffic-Cascade to identify, analyze, and visualize traffic cascades using publicly available bus trajectory data, unveiling spatial and temporal insights of the lifecycle of congestion cascades. Traffic-Cascade allows general public and urban traffic controller to view, explore, and inspect traffic health status. In the future, we plan to enhance the system in order to mine and visualize data in real time.

[4]https://www.elastic.co/
[5]https://nodejs.org/

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

REFERENCES

[1] Tarique Anwar, Chengfei Liu, Hai L. Vu, and Md. Saiful Islam. 2016. Tracking the Evolution of Congestion in Dynamic Urban Road Networks. In *CIKM*.
[2] Po-Ta Chen, Feng Chen, and Zhen Qian. 2014. Road Traffic Congestion Monitoring in Social Media with Hinge-Loss Markov Random Fields. In *IEEE ICDM*.
[3] Meng-Fen Chiang, Ee-Peng Lim, Wang-Chien Lee, and Agus Trisnajaya Kwee. 2017. BTCI: a New Framework for Identifying Congestion Cascades Using Bus Trajectory Data. In *IEEE Big Data*.
[4] Constantinos Costa, Georgios Chatzimilioudis, Demetrios Zeinalipour-Yazti, and Mohamed F. Mokbel. 2017. Towards Real-Time Road Traffic Analytics Using Telco Big Data. In *BIRTE*.
[5] Eleonora D'Andrea, Pietro Ducange, Beatrice Lazzerini, and Francesco Marcelloni. 2015. Real-Time Detection of Traffic From Twitter Stream Analysis. *IEEE Transactions on Intelligent Transportation Systems* 16 (2015), 2269–2283.
[6] Meiling Liu, Kaiqun Fu, Chang-Tien Lu, Guangsheng Chen, and Huiqiang Wang. 2014. A search and summary application for traffic events detection based on Twitter data. In *SIGSPATIAL/GIS*.
[7] Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics* 33, 3 (1962), 1065–1076.
[8] Yizhou Sun, Charu C. Aggarwal, and Jiawei Han. 2012. Relation Strength-aware Clustering of Heterogeneous Information Networks with Incomplete Attributes. *Proc. VLDB Endow.* 5, 5 (Jan. 2012), 394–405.
[9] Senzhang Wang, Lifang He, Leon Stenneth, Philip S. Yu, and Zhoujun Li. 2015. Citywide traffic congestion estimation with social media. In *SIGSPATIAL/GIS*.
[10] Senzhang Wang, Xiaoming Zhang, Jianping Cao, Lifang He, Leon Stenneth, Philip S. Yu, Zhoujun Li, and Zhiqiu Huang. 2017. Computing Urban Traffic Congestions by Incorporating Sparse GPS Probe Data and Social Media Data. *ACM Trans. Inf. Syst.* 35 (2017), 40:1–40:30.
[11] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. v. d. Wetering. 2013. Visual Traffic Jam Analysis Based on Trajectory Data. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2159–2168.