Intelligent Traffic Analytics: From Monitoring to Controlling

Sheng Wang, Yunzhuang Shen, Zhifeng Bao
School of Science
RMIT University
firstname.surname@rmit.edu.au

Xiaolin Qin
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
qinxcs@nuaa.edu.cn

ABSTRACT

In this paper, we would like to demonstrate an intelligent traffic analytics system called T4, which enables intelligent analytics over real-time and historical trajectories from vehicles. At the front end, we visualize the current traffic flow and result trajectories of different types of queries, as well as the histograms of traffic flow and traffic lights. At the back end, T4 is able to support multiple types of common queries over trajectories, with compact storage, efficient index and fast pruning algorithms. The output of those queries can be used for further monitoring and analytics purposes. Moreover, we train the deep models for traffic flow prediction and traffic light control to reduce traffic congestion. A preliminary version of T4 is available at https://sites.google.com/site/shengwangcs/torch.

ACM Reference Format:
Sheng Wang, Yunzhuang Shen, Zhifeng Bao and Xiaolin Qin. 2019. Intelligent Traffic Analytics: From Monitoring to Controlling. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3289600.3290615

1 INTRODUCTION

Traffic congestion has always been a main roadblock of urban commuting, and it was reported that traffic congestion costs Americans $124 billion a year [11]. Existing solutions for solving traffic congestion still highly depends on large amount of human resource to monitor the CCTVs collected from cameras and sensors in the roads, manually control the traffic lights with fixed timetable if serious congestion occurs, or send police labors to direct the traffic. With more and more vehicles on the road, human control becomes more difficult to conduct in real time.

In this paper, we demonstrate an intelligent traffic analytics system over the road network to assist the traffic monitoring and control, T4 — which is supported by four modules. In the first two modules, our latest proposed trajectory search engine-Torch [10] and an interactive visualization platform support efficient and effective explorations over real-time and historical trajectories. The trajectory data of vehicles in the road network collected from the cameras and GPS devices are first modeled and imported to our search engine. We visualize the real-time traffic flow by building histogram for streets and conducting clustering over the streaming trajectory data. To mine the pattern of historical trajectories, T4 enables users to interactively discover the trajectories by searching, such as range query, path query, k nearest neighbor query and reverse k nearest neighbor query [9]. Based on the automatically mined patterns and accumulated volume records of each road, a deep traffic prediction model is trained to conduct the volume prediction of each monitored road. The predicted result will be further transformed to the traffic light control model training. As a result, we aim to reduce the traffic congestion by exploiting T4.

T4 is a system that integrates the traffic visualization, monitoring, prediction and controlling, and all these analytic tools are powered by our fast search engine Torch [10]. The code of our back-end search engine has been open sourced[1]. A preliminary version of the system over several real-world trajectory datasets is online accessible[2], the first two modules are fully implemented, and we will keep optimizing the last two modules with new models.

In the traffic visualization module, instead of showing all individual trajectories' movement which is overwhelming, we present the real-time traffic flow by a fast clustering algorithm, as well as the frequency flow of streets with a histogram. In the traffic monitoring, based on trajectory search [2, 4], we support: searching by a region, path, street name or trajectory. Time stamp can be coupled with each of the above queries to form a spatial-temporal counterpart. All together they form a comprehensive pool of query types that are commonly used by traffic officers for monitoring and decision making. In the traffic prediction module, we keep training a deep model with the latest statistics reported by our search engine instead of using the static dataset and model. In the traffic light control module [11], we optimize the traffic light control based on the predicted traffic volume of each road at city-level, rather than focusing only on regional and simulation level optimization.

Besides, T4 can also be used for bus or metro route planning to increase the ridership of public transportation [9], and site selection of facility deployment [14] to maximize the coverage of facilities at urban level. Specifically, analyzing the taxi trajectory dataset by a reverse k nearest neighbor query over trajectories [9] can help the public transportation department estimate the demand of people's commuting on the newly planed route, and further reduce the greenhouse gas emission.

2 SYSTEM ARCHITECTURE

As shown in Figure 1, T4 includes four modules, a front end (green) visual interface (Travis) and a back end (light blue) of search processing and deep model training for traffic prediction and control.

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3290615

[1]https://github.com/tgbnhy/torch-trajectory
[2]https://sites.google.com/site/shengwangcs/torch

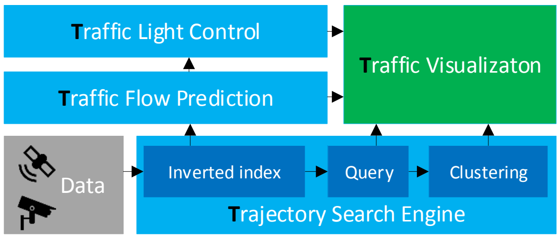

Figure 1: System architecture of T4.

In the back end, the query and clustering requests are handled efficiently supported by our search engine [10], which keeps modeling the streaming traffic records (grey) collected from the cameras in the main roads of city, and updates the inverted index. In the front end, T4 presents traffic information to users in a visually effective way (Figure 2). It shows the traffic flows by an efficient clustering algorithm over streaming trajectory data, and the frequency histogram of the streets, a search panel enables user to discover the historical and real-time trajectory in a range, a path in the map, or conduct the k nearest neighbor search and reverse k nearest neighbor query. The deep models for traffic volume prediction (Trap) and light control (Tralcon) are continuously being trained based on the summarized frequency of each road from our search engine, to assist the light control model in reducing congestion once identified.

3 BACK-END TECHNIQUES

In this section we first describe how to model the data collected from vehicles and how the search engine works. For details, we refer readers to our work [10]. Then we briefly introduce the deep models we use for traffic prediction and traffic light control.

3.1 Data Modeling

DEFINITION 1. ***(Road Network)*** *A road network is a graph* $G = (V, E)$*, where* V *is a set of vertices* v *representing the intersections and terminal points of the road segments, and* E *is a set of edges* e *representing road segments.*

DEFINITION 2. ***(Path)*** *A path* P *is composed by a set of connected road segments* $e_1 \rightarrow e_2 \rightarrow \ldots \rightarrow e_{n+1}$ *in* G*.*

Both camera trajectories and GPS trajectories composed by points can be mapped to paths in the road network easily. We use a well-adopted map matching method [6] to project the raw GPS trajectory T to the road network path P. For the trajectory collected by the cameras, it is even easier to map as the cameras are always deployed at the vertex of the road network.

DEFINITION 3. ***(Edge Inverted Index)*** *The inverted index* I_e *of an edge* e *stores the tuples* {*traID, order*} *of all map-matched segment-trajectories* T *that overlap with* e*, where traID is the unique identifier of* T *and order is the position of the edge* e *in* T*.*

3.2 Trajectory Search Engine-Torch

Multiple Query Types. According to the user input in Table 1, we support multiple query types for user to explore the historical trajectories. They can be broadly divided into two groups. The first one is for the purpose of monitoring, i.e., searching specific trajectories by street names or regions, the query can be coupled with timestamps. The second one is for the purpose of advanced analytics, such as the k nearest neighbor (kNN) search and reverse k nearest neighbor search over trajectories (RkNNT) [9] which will be used to estimate the traffic flow, and clustering can work for capturing the traffic trend.

Table 1: An overview of supported queries in the search engine. For the distance measure in kNN, we support six different measures.

Scenario	Query type	User Input	Example
Monitoring	Path Query	Path	–
	Range Query	Square or Circle	□ , ○
	Street Query	Street name	"Wall Street"
Advanced Analytic	kNN	k, a trajectory, Distance measure	Similar trips
	RkNNT	k, a trajectory	Ridership prediction
	Clustering	k, trajectories	Representative paths

Compact Index and Storage. We build a lightweight edge & vertex index (LEVI). As the inverted index is composed of posting lists of trajectory identifiers (traID), denoted by integers, we further compress the sorted list of integers using the *delta encoding* [3]. Timestamp and order information is also stored. Furthermore, we exploit the *variable byte* technique to further compress the trajectory data, which is essentially an array list of unsorted integers.

Dynamic Pruning. Similar to dynamic pruning strategies such as MAXSCORE [7] over posting lists from the Information Retrieval area, and the *Threshold algorithm* [1] from the database area, our algorithm is composed of filtering and refinement. To filter out non-qualified trajectories, the inverted indexes play an important role in every similarity measure. Termination occurs when the k result in the result set has a similarity greater than the upper bound for the remaining unprocessed trajectories. The refinement step is based on bound reordering techniques, such as sorting all of the candidate trajectories by upper bound similarity, then computing the true similarity through pivoting. Processing stops when the next upper bound is smaller than the k result.

3.3 Traffic Prediction-Trap

Based on our edge inverted index divided by timestamp, we can collect the volume in each time stamp easily and use it as the training data to predict future traffic volume. A common input of the deep prediction model is to construct a matrix $\mathcal{M}_t$ at timestamp t based on the road network, i.e., $\mathcal{M}_t[i][j]$ denotes the volume from vertex i to j. The prediction problem can be illustrated as follows: given the historical observations $\{\mathcal{M}_t | t = 0, \cdots, n-1\}$, predict $\mathcal{M}_n$. A sequence of such matrices are used to train with a deep model which combines the convolutional neural network (CNN) and recurrent neural network (RNN). We support multiple such deep models to predict the traffic volume $\mathcal{M}_n$, such as STResNET [13], DCRNN [5]. We also plan to build a deep model which integrates the real-time traffic flow to the training, i.e., new records$\{\mathcal{M}_t | t = n, \cdots\}$ will be fed into the training process.

3.4 Traffic Light Control-Tralcon

The above predicted volume will indicate the future status of each road intersection, which can be further used to optimize the signal in advance to reduce the congestion. We use deep reinforcement learning [8, 11] to optimize the traffic lights. Specifically, at each road intersection, a *traffic light control agent* makes decisions based on the *multiple statistics* (e.g. current traffic volume, speed on related roads provided by our search engine Torch in real time, and the future volume predicted by Trap in specific regions). Inside each agent, a deep neural network which maps the current observation to an *action* (i.e., alter the light state and corresponding duration), is trained based on the *reward* defined as a composition of feedback (i.e. queue length, waiting time, lane speed).

4 DEMONSTRATION SCENARIOS OF T4

T4 currently supports three real-world datasets. Two of them are the taxi datasets of Beijing [12] and Porto[3], where each trajectory is collected from GPS equipped in the taxis. We projected them on the road network by map matching, and the mapped datasets are available online[4]. Another dataset is the real traffic camera record dataset of a median city in China. There are 649 cameras in the main roads, which collected nearly 0.3 billions records in 31 days. In this paper, we will mainly use the Porto dataset to demonstrate different use cases due to page limit.

4.1 An Overview of Interface

As shown in Figure 2, Travis has a central map with three panels. The map enables users to form the queries by drawing squares or lines, whose results will be presented on the map. On the left side, the search panel enables users to specify the query types and parameters powered by Torch. On the upper right position, the traffic prediction panel will show the prediction of traffic volume of selected road via histogram. The histogram contains historical frequency for previous time slots as well as the prediction conducted on Trap for future time slots. On the lower right position, the traffic signal panel will dynamically show the state of selected traffic light over certain time span and display relevant statistics for nearby roads. The central map and three panels will be used together to demonstrate the capability of T4. Below we will describe four scenarios to further elaborate how T4 works when a user Grace wants to explore the historical and future traffic of Porto.

4.2 Scenario 1: Trajectory Search

To search the trajectories across a specific region or through a path, Grace can construct a query by plotting a window or a path on the map directly. Torch will recognize the query type concluded in Table 1, and record the query into search panel that shows up on the left side. In the search panel, Grace can further add the temporal constraint over the query or specify the value of k and the distance measure for the kNN search.

T4 also enables Grace to form a combined query by selecting multiple queries at the same time from the search panel. After the query is processed and the qualified trajectories are retrieved, T4 can further visualize them in different modes, which will be further elaborated in next scenario. Figure 3 shows an exemplar query that consists of two window queries.

4.3 Scenario 2: Trajectory Visualization

There can be hundreds or thousands qualified trajectories for some queries, and displaying all of them at once is impractical and overwhelming. Travis provides two ways to explore the large result set, which are via *scrollable list* and *clustering* respectively. For the first approach, the trajectories are loaded into a scrollable list view on the search panel. The results are divided into small chunks and displayed on the map in batch while the user scrolls down the list. Grace also can view each trajectory while the others are hidden automatically. Figure 3 shows the results using the scrollable list. Further, if Grace wants to have an overview of the result set, an efficient clustering algorithm can be conducted by T4 to generate k representative trajectories, which can also be specified by Grace. Figure 4 shows the results produced by our clustering algorithm. These two techniques make the trajectory visualization more practical and acceptable for the purpose of traffic flow analysis.

4.4 Scenario 3: Traffic Prediction

Grace can further explore the traffic prediction of each monitored road. To observe the change of traffic volume on a specific road, Grace can directly click on the monitored road on the map. The statistics of the road will be shown on the traffic prediction panel. As shown in Figure 2(d), the horizontal axis of the histogram is the time slot and the vertical axis is the number of cars passing the road within the time slot. The bars in green shows the current traffic volume, while the bars in grey show the historical frequency in previous time slots, and the bars in blue represent the traffic prediction for future time slots powered by Trap. We will also prompt Grace by changing the color of the road to red when the estimated volume of that road exceeds the volume threshold. The threshold is learned from the historical dataset.

4.5 Scenario 4: Traffic Light Control

Grace can also observe the historical and future traffic light states predicted based on Tralcon. The traffic light can be specified by clicking at the intersection area of two roads, and the statistics will be presented in Figure 2(e). The green line indicates the current traffic light state, and the grey line means the historical traffic light and the line in blue means the future state of the traffic light. Grace can have a hint of the traffic light state by watching the traffic volume of those intersected roads below the histogram. Further, Grace can also change the traffic light state manually to monitor the simulated traffic flow, and help her make a proper decision to direct the real traffic in advance.

ACKNOWLEDGMENTS

This work was partially supported by ARC DP170102726, DP180102050, and NSFC 61728204, 91646204. Zhifeng Bao is a recipient of Google Faculty Award.

[3] http://www.geolink.pt/ecmlpkdd2015-challenge/whoweare.html
[4] https://sites.google.com/site/shengwangcs/torch

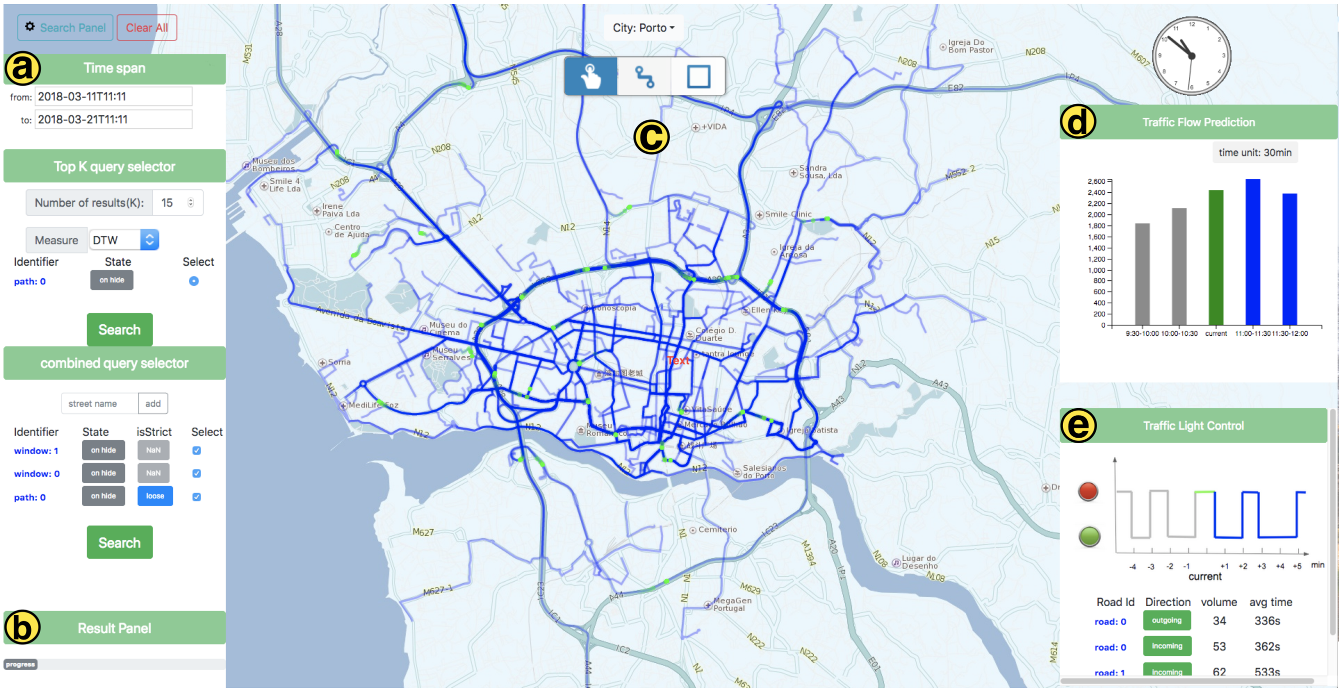

Figure 2: An overview of the main interface of T4: (a) the search panel, which allows users to specify the spatial, temporal constraints and other query configurations; (b) The list view that displays the id of each retrieved trajectory; (c) the map that enables the user to construct queries; (d) the traffic prediction panel for displaying the historical data and predicting future traffic volume; (e) the traffic light control panel for displaying the change of signal states of selected traffic lights over time.

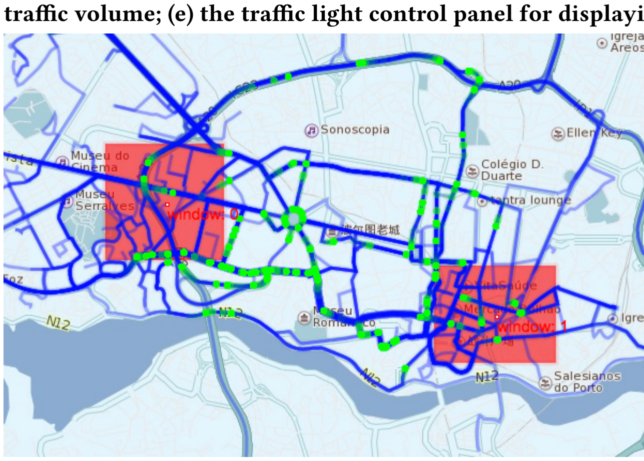

Figure 3: Combined range queries and the returned trajectories crossing the two query windows simultaneously.

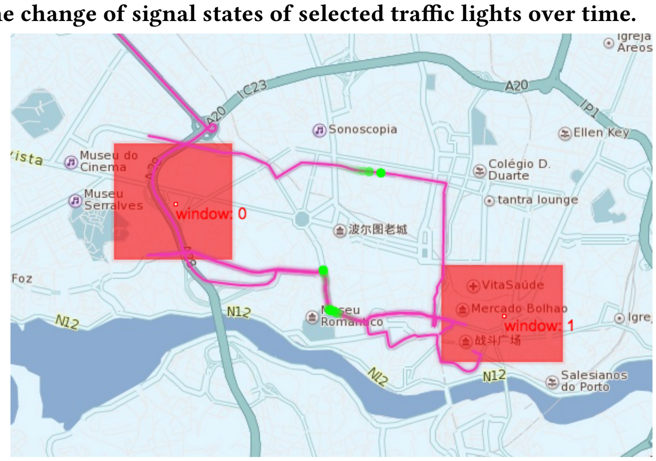

Figure 4: Four representative trajectories returned by the clustering algorithm over the result trajectories in Figure 3.

REFERENCES

[1] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences*, 66(4):614–656, 2003.

[2] B. Krogh, O. Andersen, E. Lewis-Kelham, N. Pelekis, Y. Theodoridis, and K. Torp. Trajectory based traffic analysis. In *SIGSPAIAL*, pages 526–529, 2013.

[3] D. Lemire and L. Boytsov. Decoding billions of integers per second through vectorization. *Software: Practice and Experience*, 45(1):1–29, 2015.

[4] R. Li, S. Ruan, J. Bao, and Y. Zheng. A Cloud-Based Trajectory Data Management System. In *SIGSPAIAL*, pages 3–6, 2017.

[5] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*, 2018.

[6] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-matching for low-sampling-rate GPS trajectories. In *SIGSPATIAL*, pages 352–361, 2009.

[7] H. Turtle and J. Flood. Query evaluation: strategies and optimizations. *Inf. Process. Manage.*, 31(6):831–850, 1995.

[8] E. Van Der Pol and F. A. Oliehoek. Coordinated Deep Reinforcement Learners for Traffic Light Control. In *NIPS*, 2016.

[9] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, and G. Cong. Reverse k nearest neighbor search over trajectory. *IEEE Transactions on Knowledge and Data Engineering*, 30 (4):757 – 771, 2018.

[10] S. Wang, Z. Bao, J. S. Culpepper, Z. Xie, Q. Liu, and X. Qin. Torch : A Search Engine for Trajectory Data. In *SIGIR*, pages 535–544, 2018.

[11] H. Wei, G. Zheng, H. Yao, and Z. Li. IntelliLight : a Reinforcement Learning Approach for Intelligent Traffic Light Control. In *KDD*, pages 2496–2505, 2018.

[12] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-Drive : driving directions based on taxi trajectories. In *GIS*, pages 99–108, 2010.

[13] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*, pages 1655–1661, 2017.

[14] P. Zhang, Z. Bao, Y. Li, G. Li, Y. Zhang, and Z. Peng. Trajectory-driven Influential Billboard Placement. In *KDD*, pages 2748–2757, 2018.