A Machine Learning Approach to Find the Optimal Routes Through Analysis of GPS Traces of Mobile City Traffic

Shreya Ghosh, Abhisek Chowdhury and Soumya K. Ghosh

Abstract The rapid urbanization in developing countries has modernized people's lives in various aspects but also triggered many challenges, namely increasing carbon footprints/pollution, traffic congestion and high energy consumption. Traffic congestion is one of the major issues in any big city which has huge negative impacts, like wastage of productive time, longer travel time and more fuel consumption. In this paper, we aim to analyse GPS trajectories and analyse it to summarize the traffic flow patterns and detect probable traffic congestion. To have a feasible solution of the traffic congestion issue, we partition the complete region of interest (ROI) based on both traffic flow data and underlying structure of the road network. Our proposed framework combines various road features and GPS footprints, analyses the density of the traffic at each region, generates the road-segment graph along with the edge-weights and computes congestion ranks of the routes which in turn helps to identify optimal routes of a given source and destination point. Experimentation has been carried out using the GPS trajectories (T-drive data set of Microsoft) generated by 10,357 taxis covering 9 million kilometres and underlying road network extracted from OSM to show the effectiveness of the framework.

Keywords GPS trace · OpenStreetMap (OSM) · Classification · Traffic

1 Introduction

The staggering growth of urban areas and increased populations drive governments for the improvement of public transportation and services within cities to reduce

S. Ghosh (✉) · A. Chowdhury · S. K. Ghosh
Department of Computer Science and Engineering, Indian Institute of Technology,
Kharagpur, India
e-mail: shreya.cst@gmail.com

A. Chowdhury
e-mail: abhisekchowdhury9@gmail.com

S. K. Ghosh
e-mail: skg@iitkgp.ac.in

P. K. Sa et al. (eds.), *Recent Findings in Intelligent Computing Techniques*,
Advances in Intelligent Systems and Computing 708,
https://doi.org/10.1007/978-981-10-8636-6_7

carbon footprints and make a sustainable urbanization [1, 2]. As traffic congestion poses a major threat to all growing or developing urban areas, analysing traffic flows and detecting such issues in urban areas are strategically important for the improvement of people's lives, city operation systems and environment. Noticeably, with the availability of GPS data and advances in big data and growing computing and storage power, urban planning, capturing city-wide dynamics and intelligent traffic decisions have attracted fair research attentions in last few decades. These lead to various interesting applications in navigation systems and map services like route prediction, traffic analysis and efficient public transportation facilitating people's lives and serving the city [3, 4]. Also with the advent of emerging technologies like self-driving car, we need innovative technologies and an efficient and fault-tolerant traffic management system which can automatically and unobtrusively monitor traffic flow and detect any anomaly condition like traffic congestion and road blockage. Prior research on GPS traces analysis [5], traffic flow analysis and prediction has focused on the analysis of individual's movement patterns [4]. Few works have studied traffic flow conditions in a city region [6]. But, traffic flow prediction from individual level becomes difficult due to individual's life patterns and randomness of human movement nature. To this end, we have addressed the problem of traffic congestion detection from the city-wide vehicular movements and also analysed various road network features which were missing in the existing studies. We turn our attention to design a smart and automatic system that will detect the congestion in real time and subsequently manage it efficiently to ensure smooth traffic flow. To address the aforementioned traffic congestion problems, we propose a framework which involves (i) segmentation of road network and creating buffered regions of different road types, (ii) generating trajectory density function at each road type and analysing probability distribution of traffic flow, (iii) selection of traffic congestion features of a region and threshold depiction and (iv) build road-segment graph of the region and analyse congestion ranking of the paths. One of the real-life application scenario is rerouting of vehicles to avoid more traffic problems or even a recommendation system can be built which will notify the traffic condition of the major intersection points of a road network to carry out intelligent routing decisions. The remainder of the paper is organized as follows: Section 2 illustrates our proposed congestion detection and management framework; Section 3 describes the basic features of the visualization model and represents the obtained result; and finally, Sect. 4 concludes the paper with a highlight on the scope of future work.

2 Architecture of the Proposed Framework

In this section, we present the overall architecture of the proposed framework. Figure 1 depicts various modules of the framework. *GPS data pre-processing* module involves extracting road map of the region of interest (ROI) along with the OpenStreetMap (OSM) road features of the region. To analyse large-scale traffic data of any city region, a systematic storage method of GPS data with all contextual

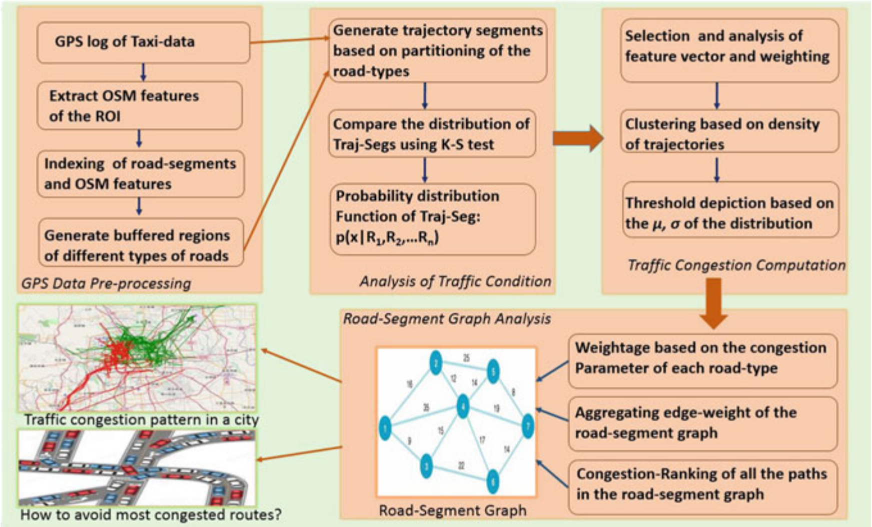

GPS log of Taxi-data
Extract OSM features of the ROI
Indexing of road-segments and OSM features
Generate buffered regions of different types of roads
GPS Data Pre-processing
Generate trajectory segments based on partitioning of the road-types
Compare the distribution of Traj-Segs using K-S test
Probability distribution Function of Traj-Seg: $p(x|R_1,R_2,\ldots R_n)$
Analysis of Traffic Condition
Selection and analysis of feature vector and weighting
Clustering based on density of trajectories
Threshold depiction based on the μ, σ of the distribution
Traffic Congestion Computation
Road-Segment Graph Analysis
Weightage based on the congestion Parameter of each road-type
Aggregating edge-weight of the road-segment graph
Congestion-Ranking of all the paths in the road-segment graph
Road-Segment Graph
Traffic congestion pattern in a city
How to avoid most congested routes?

Fig. 1 Architecture of the framework

information and road network is required. To this end, we used a storage schema to efficiently access useful information for our application. The *Analysis of Traffic Condition* and *Traffic Congestion Computation* modules generate trajectory segments from the raw GPS log of the vehicles and compute the probability density function of the GPS footprints in each of the road types. Based on the computed density function of the historical GPS log, threshold parameter of the congestion in each type of roads is determined. In the next module, namely *Road-Segment Graph Analysis*, road graph is modelled and edge-weights are assigned. Based on the congestion ranking of each route, a probable less-congested path from a given source and destination can be identified. The framework also summarizes the traffic congestion pattern of the region of interest from the GPS footprints and road-feature set. The details of the modules are described in the following sections of the paper.

2.1 GPS Data Pre-processing

We aim to extract optimal routes analysing GPS traces of the city traffic. A typical GPS log consists of a series of timestamped latitude and longitude: $< lat_i, lon_i, t_i >$. To tackle the traffic congestion issues, the timestamped positions of traffic are not sufficient, underlying road network and road features play an important role. Therefore, we extract underlying road network of the region of interest from OpenStreetMap (OSM) [7]. From the GPS data log, we create the bounding box which covers all GPS traces and extract the road network (.shp file) from the OSM map.

To bridge the gap between extracted road map and GPS log, we use *Map-matching* technique, which aligns the sequence of GPS log points with the underlying road structure on a digital map. We have implemented *ST-Matching* algorithm [8] which captures spatial geometric, topological structures as well as the temporal or speed limitation of the GPS traces. The output of the algorithm is set of <lat, lon> (latitude, longitude) sequences along with the unique road-ID from OSM. After the fundamental pre-processing step, we need to append and use the road-feature information from the unique OSM ID which represents a set of road features of the particular road segment. The OSM-key is used to identify any kind of road, street or path, and it indicates the importance of the road within the road network of the region. We have extracted the key value of the road segments and created a buffered region for all the road segments. Table 1 depicts various types of roads and the corresponding buffered regions taken in our experimental set-up. After appending the contextual information with the GPS log and road network, we store the buffered regions based on the spatial bounding of each region.

Table 1 Road network feature and buffered regions

OSM-key/value	Road network feature	Buffered region
Motorway	Major divided highway generally having two or more running lanes and emergency hard shoulder	$l_w + 3.5$ km
Trunk	The most important roads in a country's system and may not have a divided highway	$l_w + 3$ km
Primary	A major highway which links large towns, in developed countries normally with two lanes	$l_w + 2.5$ km
Secondary	A highway which forms linkage in the national route network.	$l_w + 2$ km
Tertiary	It connects or links between smaller settlements, and local centres also connects minor streets to major roads	$l_w + 1$ km
Residential	Roads accessing or around residential areas, street or road generally used for local traffic within settlement	$l_w + 1$ km
Service	Access to a building, service station, beach, campsite, industrial estate, business park, etc.	$l_w + 0.5$ km

2.2 Analysis of Traffic Condition

Road segmentation (i.e. classification of various road segments in the data set to ‘highway’ or ‘local’) is followed by feature selection. Feature selection or extraction is one of the most important for selecting a subset of relevant features for the construction of a predictive model. In our problem, probable traffic congestion is determined by average velocity and number of vehicles in a particular bounding box or region or increased vehicle queuing. It has been considered that congestion may occur if the number of cars in a bounding box is greater than a particular threshold value and the average velocity of cars is less than a particular threshold value. Threshold values vary for different types of buffered regions like highways and locality region. The challenges are to predict or detect possible congestion from traffic flow. In addition, different types of buffered regions may have different parameters for traffic condition detection due to the different capacities of the regions. Traffic congestion (T_c) can be represented as in Eq. 1, where v, n, l_w are average velocity, number of vehicles and width of the road, respectively. α and β are normalizing constants.

$$T_c = \frac{\alpha}{v} + \frac{n}{l_w} \times \beta \tag{1}$$

To model traffic flow of a region, we need to summarize road conditions in different time interval. It is quite obvious that traffic scenario at morning(5.00–7.00 am) will significantly differ in peak times (9.00–11.00 am). Therefore, we need to model the time-series data $x = (x_0, x_1, \dots, x_n)$, where each x_i represents T_c value at t_i timestamp. We divide each day in equal partition of 1 h and carry out the analysis. After plotting the histogram of the above time-series data, we observe that Gaussian distribution can approximate the histograms. Using parametric method of distribution fitting, we estimate the μ and σ^2 of the distribution from the mean and standard deviation of the data set, where mean, $m = \frac{\sum T_c}{t_n}$; standard deviation, $s = \sqrt{\frac{1}{t_n-1} \times \sum(T_c - m)}$. In this section, we demonstrate the process of analysing GPS log along with road features and summarize the traffic movements by defining a probability distribution for seven different regions (depicted in Table 1) of the road network.

2.3 Traffic Congestion Computation

For the traffic congestion computation, two sub-modules are required. First, we need to generate the normal traffic probability distribution from the historical GPS log; next, we need to carry out a classification algorithm for classifying the regions into congested and non-congested regions. Traffic congestion on a road segment means disrupting the normal traffic flow of the region. Hence, to detect such condition we depict congestion threshold parameter which will denote probable traffic bottleneck of fluctuations in normal traffic conditions. We use *k-nearest neighbour* classifier for this learning. It is based on learning by analogy, i.e. it compares a given test tuple with other training tuples having similar feature set. The training tuples are described by a set of feature attributes. *'Closeness'* or *'Similarity'* is defined in terms of a distance or similarity metric. We have used Euclidean distance measure given two points or tuples, say, $X_1 = (x_{11}, x_{12}, x_{13} \dots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, x_{23} \dots, x_{2n})$, is $dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$ where each entry of the tuple represents the feature values of the elements denoting the road traffic condition at different timestamp values. We use $k = 3$, where neighbourhood is defined based on the road network feature ranking in Table 1. Algorithm 1 provides an abstract view of the process, where based on the test data set (road segments), road-feature rank is extracted and neighbourhood is defined. Then, based on the k-NN classifier algorithm, we classify the test data set by comparing it with training tuples. Here, we specify p = 5, i.e. congestion ranking is carried out, where p = 1 denotes no congestion, and p = 5 denotes highest level congestion.

2.4 Road-Segment Graph Analysis

Road-Segment Graph: $RGraph = \{(V, E) | 1 < v_i < |V|, 1 < e_i < |E|\}$, where each node $v_i \in |V|$ denotes road intersection point of the underlying road network, and each edge of the graph $e_i \in |E|$ denotes existing roads between two or more intersecting points. Each node $v_i \in |V|$ stores information about the intersection point: [*ID*, *Name*, *CLevel*]. CLevel represents congestion level at a particular intersection point which is derived from the data distribution of the connected edges. It is observed that traffic congestion normally follows a regular pattern which can be detected from the traffic density function and incorporated in the *RGraph* structure. We use a cost function to feed in the traditional shortest path algorithm. One of the most popular and cost efficient methods to determine single source shortest path is Dijkstra Algorithm. Specifically, the cost function is

$$f(n) = g(n) + h(n) \tag{2}$$

The algorithm finds out the value of g(n), i.e. the cost of reaching from initial node (entry point) to n. A heuristic estimate (function h(n)) is carried out to reach from on to any goal node or exit points. In our approach, g(n) is calculated by aggregating edge-weights of road network and h(n) is determined by calculating traffic congestion data.

Algorithm 1: Traffic condition detection based on density analysis and congestion threshold parameter

```
Input: Traffic density function of various regions: $\mathcal{N}(\mu, \sigma|R)$, congestion threshold
       parameter: Thres
Output: Road-segment Graph: RGraph, Congestion level: C
initialization: Create $RGraph(V, E)$ :V represents intersection of road-links, E: road
segments of the network;
while $e_i \in E$ do
    r=ExtractRoadType($eId_i$);
    for all training data set available do
        T=Extract $f(x|r-1, r, r+1)$;
    end
    for all time intervals n in the traffic time series and $t \in T$ do
        $p_i = f(x|n_i, r)$ ;
        $s = compareTraffic(p_i, f(x|n_i, t))$ ;
        if $s > Thresh$ then
            congest=1;
            $C$=(s-Thresh)/p;
            Append($C, RGraph$);
        else
            congest=0;
            $C$=1 ;
            Append($C, RGraph$);
        end
    end
end
```

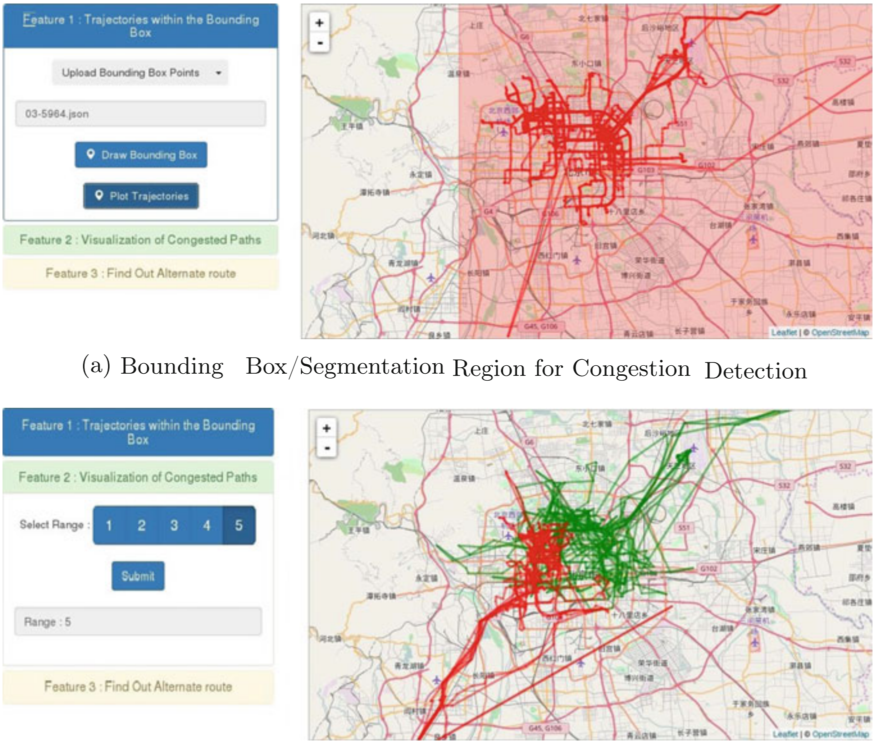
3 Experimentation

3.1 Data set

Beijing T-drive Data Set: The data set contains the GPS trajectories of 10,357 taxis during the period of 2 February–8 February 2008 within Beijing. The data set contains huge amount of GPS points, about 15 million and a total distance covered by the trajectories reaches to 9 million kilometres [3, 4, 9].
OSM Map: OpenStreetMap stores physical features on the ground (e.g. roads or buildings) using OSM-tags attached to its basic data structures (its nodes, ways and relations). Each tag depicts a geographic attribute of the feature being represented by that specific node, way or relation [7].

3.2 Results and Discussion

After the data pre-processing steps and threshold depiction steps are completed, we partition the data set in ‘Training Data Set’ and the remaining as ‘Test Data Set’. In our experiment, 1195 examples constituted the ‘Training Data Set’. The learning model is trained using the concept of k-nearest neighbours with training examples from the ‘Training Data Set’. Specifically, model is run for 25 iterations and accuracy of the model (in terms of % of examples correctly classified) is calculated. Overall accuracy of the model is 93%. Our visualization model, developed with PHP, Python, JQuery and AJAX, imitates traffic congestion on a predefined road map of the ROI. We have used OSM Leaflet API to simulate the map. On the back-end system, we have stored the data in PostgreSQL database and retrieve the data using spatial extension of psql. The model has salient three features: Feature I performs the pre-processing steps, including generation of bounding box and extracting road features and partitioning the road network into various buffered regions. Feature II depicts the level of congestion (traffic flow condition) in different regions of the network. In Feature III, the system recommends alternative routes based on the traffic flow condition on the road. Few visualization results are shown in Fig. 2.


Feature 1 : Trajectories within the Bounding Box
Upload Bounding Box Points
Draw Bounding Box
Plot Trajectories
Feature 2 : Visualization of Congested Paths
Feature 3 : Find Out Alternate route
Select Range : 1 2 3 4 5
Submit
Leaflet | © OpenStreetMap

03-5964.json

Range : 5

(a) Bounding Box/Segmentation Region for Congestion Detection

(b) Visualization of congestion Detection

Fig. 2 Snapshots of the visualization framework

4 Conclusion and Future Work

In this paper, we aim to detect optimal route in a city analysing the city-wide traffic flow. It involves segmentation of road network into different types of roads and analysing traffic condition in each of them. It models the regular traffic patterns in a city region from the mobile traffic data and detects anomalous conditions like fluctuation of road traffic from the data. Finally, the model classifies traffic condition in different categories of congestion level and few visualization results are shown. In future, we would like to build a real-time recommendation system, which can adaptively recommend the optimal route to the users. Further, we would like to enhance the accuracy and efficiency of the proposed model by using different indexing or storage schema and deploying advanced machine learning techniques.

References

1. Zheng, Y., et al.: Urban computing with taxicabs. In: Proceedings of the 13th International Conference on Ubiquitous Computing. ACM (2011)
2. Zheng, Y., et al.: Urban computing: concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol. (TIST) **5**(3), 38 (2014)
3. Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: driving directions based on taxi trajectories. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS'10, New York, NY, USA, pp. 99–108. ACM (2010)
4. Zheng, Y., et al.: Learning transportation mode from raw gps data for geographic applications on the web. In: Proceedings of the 17th International Conference on World Wide Web. ACM (2008)
5. Ghosh, S., Ghosh, S.K.: THUMP: semantic analysis on trajectory traces to explore human movement pattern. In: Proceedings of the 25th International Conference Companion on World Wide Web 2016, Montreal, Canada, Apr 11, pp. 35–36. International World Wide Web Conferences Steering Committee
6. Hoang, M.X., Zheng, Y., Singh, A.K.: FCCF: forecasting citywide crowd flows based on big data. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM (2016)
7. https://www.openstreetmap.org/
8. Lou, Y., et al.: Map-matching for low-sampling-rate GPS trajectories. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM (2009)
9. Yuan, J., Zheng, Y., Xie, X., Sun, G.: Driving with knowledge from the physical world. In: The 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11, New York, NY, USA. ACM (2011)