

# Extracting Logical Connectivity of Intersection from Floating Car Data Using Support Vector Machine Method

Min Huang<sup>1, a)</sup>, Teng Zhang<sup>1</sup>, Jiajie Pan<sup>2</sup> and Fang Liu<sup>1</sup>

<sup>1</sup>*School of Engineering//Guangdong Provincial Key Laboratory of Intelligent Transportation System, Sun Yat-sen University, Guangzhou 510006, China.*

<sup>2</sup>*Shenzhen Urban Transport Planning Center, Shenzhen 518021, China.*

<sup>a)</sup> Corresponding author: huangm7@mail.sysu.edu.cn

**Abstract.** This paper aims to extract traffic rule at intersection from floating car data, traffic rule is the logical connectivity data in road network. Firstly, based on intersection data model, the geometric characteristics of intersection are presented, and the angle of arcs is calculated to get the traffic diversion. Secondly, routes are extracted from floating car data, which represent the number of turning car at intersection. Thirdly, a classification model based on support vector machine is used to classify the connected state of arcs, and the classification feature includes the number of car and traffic diversion. Finally, this method is applied in some intersections of Guangzhou Province using six days' floating car data. The experiment results reveal the accuracy is about 90 percent.

**Key words:** traffic rule extraction; connectivity; floating car data; intersection; support vector machine.

## INTRODUCTION

Urban digital map is fundamental for traffic research, providing important information of urban road network. For example, traffic planning and traffic simulation both need the basic road network and data. A complete road network should include geometric topology data and logical connectivity data. Logical connectivity means the turning accessibility between two arcs based on the traffic rule at intersection. Road extraction technology from aerial images can accurately obtain centerlines, various technologies have been developed. For example, Trinder and Wang presented a knowledge-based method for automatic extraction of road from aerial photographs and high-resolution remotely sensed imagery [1]. Other methods, level set search algorithm and mean shift clustering technique [2], reference circle and central pixel technology [3], have been applied to extract geometrical characteristics of roads with a great performance. Song and Daniel used support vector machine to classify image into two groups [4]: a road group and a non-road group. Road centerlines are extracted from the road group images which were segmented into geometrical objects. With the development of GPS technology, other convenient and efficient technologies to get road network data from floating car data are developed. Alireza and John used a new pattern recognition method to find road intersections from GPS traces [5]. Based on spatial and semantic relationships between a GPS point and its associated road segment, Li et al. classified each trajectory point, and merges every trajectory point into the candidate road network [6]. Zhang used GPS trace data to extract road network centerline based on the least squares support vector regression machine [7].

From above, most of these researches concerned more about geometric element: road centerline and intersection. However, Logical data are also important for road network. Traffic researches will be more suitable for practical application if considering turning rule, such as the optimal route problem, capacity reliability analysis and stability analysis of road network. For logical data, Chen and John used Gaussian mixture model to get lanes from GPS data [8]. Yang got turning rule by comparing lane position with GPS traces combining Gaussian mixture density model and greedy EM algorithm [9]. Zhuang proposed an experience-based method to extract turning rule from GPS data

[10]. If the number of car turning at intersection is over than an empirical value, the arcs are regarded as turning connected. His research mainly concerns about straight, turning left, right in main road and secondary road.

The objective of this paper is to present a learning-based method to extract turning rule from floating car data using support vector machine. The methodology is introduced in section 2, including introduction of intersection data model, route extraction algorithm, support vector machine for connectivity extraction. Section 3 presents a case study using six days' floating car data. The last section gives the conclusions and future research topics.

## METHODOLOGY

### Intersection Data Model

Generally, a road network can be described as  $G=(V, E)$  in classical node-arc model, where  $V= \{v_i|i=1, \dots, n\}$  is node data set,  $E=\{e_i=(v_k, v_l)| i=1, \dots, n; v_k, v_l \in V\}$  is arc data set.  $V$  and  $E$  represent the geometric elements in road network. Based on node-arc model, node-arc-angle model is introduced [11]. In this data model, a road network is  $G=(V, E, L)$ , where  $L=\{l_i=(e_k, e_l)|i=1, \dots, n; e_k, e_l \in E\}$  represent logical element. This model uses the Adjacent-arc, North-angle and Connected-state to describe the geometric and logical characteristic of intersection. North-angle means the angle between each arc and the northward direction. Connected-state denotes the turning connectivity of arcs at intersection. Figure 1 is an example intersection, and we use node-arc-angle model to describe its characteristic, as table 1 shows.

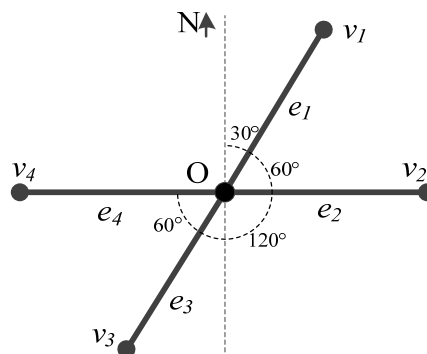


FIGURE 1. Diagram of Intersection

TABLE 1. Characteristic of Intersection

Node	Adjacent-arc	North-angle	Connected-state
O	$e_1/e_2/e_3/e_4$	30/90/210/270	0111101111011110

At this intersection, arcs of node O are  $e_1, e_2, e_3$  and  $e_4$ , the angles of arc are 30, 90, 210 and 270 degrees. The Adjacent-arc and North-angle is  $e_1/e_2/e_3/e_4$  and 30/90/210/270. If each arc is turning connected to other arcs and U-turn forbidden, the Connected-state is 0111101111011110 from  $e_1$  to  $e_4$ , where 1 represents connected and 0 is not. From North-angle, we can calculate traffic diversion as follows [12]:

$$\Delta\theta = (\theta_k - \theta_v + 360) \% 360$$

$$T = \begin{cases} 1, \Delta\theta \in [45, 135) \\ 2, \Delta\theta \in [135, 225) \\ 3, \Delta\theta \in [225, 315) \\ 4, \Delta\theta \in [0, 45) \cup [315, 360) \end{cases} \quad (1)$$

Where  $\theta_v$  is the north angle of initial arc,  $\theta_k$  is the arrival arc, and  $\Delta\theta$  is the angle between  $\theta_v$  and  $\theta_k$ . We use  $\Delta\theta$  to distinguish diversion type, where 1,2,3,4 represent left, straight, right and U-turn.

### Route Extraction

Taxis equipped with GPS receivers would send running information to the operation center in period. For example, Guangzhou has over 20,000 taxis running every day. The real-time positioning data will be transmitted to the traffic center, and over 20 million records are stored in database. Table 2 shows the row floating car data.

**TABLE 2.** Row Floating Car Data

<b>ID</b>	<b>License</b>	<b>Longitude</b>	<b>Latitude</b>	<b>Time</b>	<b>Direction</b>
26144821042	YueA1DM94	113.32147	23.14810	2014/12/15 7:00:00	80
26144821043	YueA1HA24	113.29718	23.23016	2014/12/15 7:00:20	90

A row floating car data is defined as:

$$P = \{l, lon, lat, t, d\}$$

Where  $l$  is the license number,  $lon$  and  $lat$  is longitude and latitude,  $t$  is GPS time and  $d$  is running direction. Valid positioning data would be matched with arcs in digital map by a map-matching algorithm [13-14]. The matched data is show in Table 3.

**TABLE 3.** Map Matching Results

<b>ID</b>	<b>License</b>	<b>Longitude</b>	<b>Latitude</b>	<b>Time</b>	<b>Direction</b>	<b>Arc</b>	<b>Fnode</b>	<b>Tnode</b>
26144821042	YueA1DM94	113.32147	23.14810	2014/12/15 7:00:00	80	1213	98	99
26144821043	YueA1HA24	113.29718	23.23016	2014/12/15 7:00:20	83	1213	98	99

We use  $e$  represent the matched arc,  $f_n$  and  $t_n$  represent the start node and terminated node of arc, which is distinguished by direction. So, a matched data is defined as:

$$P^m = \{l, lon, lat, t, d, e, f_n, t_n\}$$

Initial route is extracted from the map matching results based on the GPS time, and it is defined as a route sequence:

$$Tr = \{P_1^m P_2^m \dots P_i^m P_{i+1}^m \dots P_n^m\}$$

Initial Routes will be modified as follow:  $P_{mj}$  will be removed from route sequence if  $e$ ,  $f$  and  $t$  are the same as  $P_{mj-1}$ . If  $e$  is same but  $f$  and  $t$  are contrary, the car is regard as turning over and  $P_{mj}$  will be retained. In addition, redundant branch arcs, loop arcs will be removed from the route sequence. Figure 2 shows a modified route in digital map and its arcs will be stored in database.

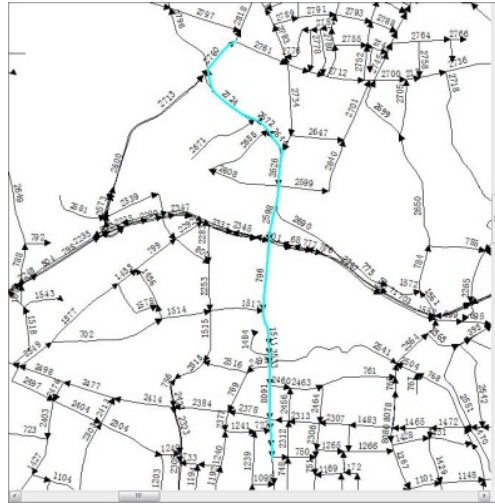


FIGURE 2. A Modified Route of Floating Car

### Support Vector Machine for Connectivity Extraction

Support vector machine is a supervised learning model, it can analyze data and recognize pattern with learning algorithms. For a given data set  $(x_i, y_i)$ , where  $x_i \in \mathbb{R}^d$ ,  $y_i \in \{+1, -1\}$ ,  $x_i$  is a  $n$ -dimensional feature vector,  $y_i$  is feature value.  $y_i$  indicates the class that  $x_i$  belongs to. The aim of classification is to find a hyperplane that divides the vector having  $y=1$  from those having  $y=-1$ . A classification is achieved by maximizing the margin between the separating hyperplane and the closet data points in the input space. This optimal hyperplane can be determined as follow,

$$\begin{aligned} \min: & \|\omega\| \\ \text{s.t.}, & y_i (\omega^T x_i + b) \geq 1, i = 1, \dots, n \end{aligned} \quad (2)$$

Solving the optimization problem, the hyperplane decision function can be written as:

$$f(x) = \text{sgn} \left\{ \sum_{i=1}^n a_i y_i (x_i \cdot x) + b \right\} \quad (3)$$

In the decision function, just a few of training points have a non-zero weight  $a_i$ . These training points lie on the margin and they are named as support vectors. This means that the decision function generated by the support vector machine is decided by the points that are closest to the hyperplane.

Based on the introduction of support vector machine, a learning-based method to extract logical data of intersection is proposed. Because  $P$  is matched to an arc, we could count the cars turning at intersection from modified route, the traffic diversion of arcs is obtained from Adjacent-arc and North- angle. The more cars turning at intersection, the more possible that those arcs are turning accessible. In addition, traffic volume is variable in different traffic diversion. For example, cars going straight are much then left, right and U-turn at some intersection, the number is higher than other diversion. So, we modeled for number of car, diversion and connected state by linear support vector machine. Connected state is  $y_i$  and it is classified into two type, we use a features vector  $x_i$  including number of car, diversion for connected state classification problem, as follows:

$$c(x) = \text{sgn} \left\{ \sum_{k=1}^m a_k c_k ((n_k, t_k) \cdot x) + b \right\} \quad (4)$$

Where  $n_k$  is the number of car,  $t_k$  is the diversion type,  $c$  is connected state. We divided the extraction task into two subtasks: training and classification. Training is involved in identifying support vectors  $S=\{(n_k,t_k)|k=1,\dots,m\}$  and weight  $A=\{a_k|k=1,\dots,m\}$  in a training dataset  $D=\{(n_i,t_i,c_i)|i=1,\dots,l\}$  of training intersections with  $c_i$  known in advance. In classification, we assigned each feature vector in test dataset  $D'=\{(n_j,t_j)|j=1,\dots,l'\}$  to  $c_j$  using formulate (4), so the connected state of two arcs is got. For feature vector, number of car is counted from modified route database, and diversion is calculated from north angle by formulate (1). However, diversion is not a numerical feature for model training. The dataset will be divided into four independent datasets by different diversion, four classification models are generated for different diversion classification, as formulate (5) shows:

$$c(x^t) = \text{sgn} \left\{ \sum_{i=1}^n a'_i c'_i (n'_i \cdot x^t) + b^t \right\}, t = 1, 2, 3, 4 \tag{5}$$

Where  $t$  represents diversion type.

The algorithm for model training and classification is shown in Fig.3.

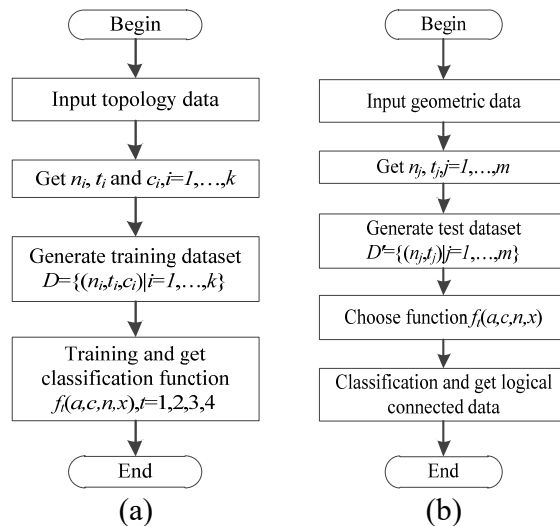


FIGURE 3. Framework of Algorithm: (a) Training; (b) Classification

### EXPERIMENTAL TESTS

In this section, we compare our method with Zhuang’s experience-based method. In his research, if the number of car is over than 30 and the percentage is over 3%, the arcs are connected. We used 5 days’ floating car data from 15 to 19 on December, 2014 for training and data in 22 December, 2014 for classification. In tests 1, 52 training intersections including main roads, secondary roads and branches are chosen from Liwan District and Yuexiu District, which include 124 arcs and 666 connected state in total. In addition, we using number of cars only to show the effect of diversion. The test results are shown in table 4. From table 3, experience-based method has a negative result for U-turn, but learning-based method has a better result. Also, considering diversion has a better result than using number of car simply, the accuracy is higher if combining number of car with diversion.

In test 2, this classification models are used to derive the connected state of intersections in Tianhe District. There are 45 arcs and 200 connected state, and the results are shown in table 5. From table 5, the performance of learning-based method is better, because the connected criteria are training from data. It is more convincible than the empirical value. The classification accuracy is higher in left turn, right turn and straight than U-turn. Because the proportion of traffic volume going straight, left and right at intersection is higher than U-turn, they are easier to

derive the connected state. In conclusion, the proposed method based on support vector machine is available for connected state classification.

**TABLE 4.** Comparison of Results for Different Method

Intersections	Diversion	Number of arcs	Accuracy of classification results		
			Experience-based method	Learning-based method	
				Without diversion	Diversion
Liwan district and Yuexiu district	Left	161	95.6%	93.7%	94.4%
	Right	161	91.3%	90.1%	92.5%
	Straight	160	97.5%	96.9%	97.5%
	U-turn	184	78.3%	84.7%	90.2%
	Subtotal	666	90.4%	91.1%	93.5%

**TABLE 5.** Test Results of Classification Models

Intersections	Diversion	Number of arcs	Accuracy of classification results	
			Experience-based method	Learning-based method
			Tianhe district	Left
Right	48	93.8%		97.9%
Straight	46	91.3%		97.8%
U-turn	58	60.3%		75.7%
Subtotal	200	83.5%		89.5%

From above, the classification error in results may be caused by follow reasons:

(1) Unavoidable errors of positioning and map-matching affect route extraction. For example, GPS data would drift because of the precision of GPS devices. There are distance errors between actual point and positioning point. Further, it is difficult to describe the road network in digital map as the real complicate road network, and the map-matching precision of GPS point is lower in viaducts or tunnels. The GPS point on viaduct or tunnel may be matched with plane road. Due to these errors, routes extracted from data may be wrong when compared with the actual trajectory. These wrong routes would affect the model training and classification results.

(2) The training dataset is important to classification model. For connected state classification, the size of training dataset should be adequate, and a significant training dataset should include various levels of road, different types of intersections and turning direction. In our research, the training intersection is chosen randomly, and we have not tested the training dataset for accuracy of classification. The training dataset may also have effect on training model and cause errors.

## CONCLUSION

To extract logical connectivity at intersections in urban road network, this paper proposed a learning-based method considering number of turning cars and traffic diversion. In test section, we compare with the experience-based method, and the results told learning-based method have a better performance. Considering diversion, the accuracy is higher than using the number of cars simply. This proposed method worked effectively in most intersections.

The limitations of this method mainly include two parts. First, we have not deal with the errors in data, which would affect route extraction. Second, we have not test the training dataset for the classification results. Further improvements to the classification model are necessary to make it adaptive and accurate. Route extraction should be developed to reduce the influence of drift error. Also, more features like road level, road compicacy should be explored and introduced to improve the classification model.

## ACKNOWLEDGEMENTS

This research described in this paper was supported by Science and Technology Project of Guangdong province (2016A020223006), National Natural Science Foundation of China (11574407).

## REFERENCES

1. Trinder, John C., and Yandong, Wang. Automatic Road Extraction from Aerial Images. *Digital Signal Processing*, Vol. 8, No. 4, 1998, pp. 215-224.
2. Revathi, M., and M. Sharmila. Automatic Road Extraction Using High Resolution Satellite Images Based on Level Set and Mean Shift Methods. *Computing, Communications and Networking Technologies (ICCCNT)*, 2013 Fourth International Conference on. IEEE, 2013, pp. 1-7.
3. Li, Yan, and Ronald Briggs. Automatic Extraction of Roads from High Resolution Aerial and Satellite Images with Heavy Noise. *World Academy of Science, Engineering and Technology*, Vol. 54, 2009, pp. 416-422.
4. Song, Mingjun, and Daniel Civco. Road Extraction Using SVM and Image Segmentation. *Photogrammetric Engineering & Remote Sensing*, Vol. 70, No. 12, 2004, pp. 1365-1371.
5. Fathi, Alireza, and John Krumm. Detecting Road Intersections from GPS Traces. *Geographic Information Science*. Springer Berlin Heidelberg, 2010, pp. 56-69.
6. Li, Jun, et al. Integrated Use of Spatial and Semantic Relationships for Extracting Road Networks from Floating Car Data. *International Journal of Applied Earth Observation and Geo-information*, Vol. 19, 2012, pp. 238-247.
7. Zhang, Liting. Road Data Updating Based on Taxi GPS Trajectory. MS thesis. Kunming University, 2014.
8. Chen, Yihua, and John Krumm. Probabilistic Modeling of Traffic Lanes from GPS Traces. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 81-88.
9. Yang, Lin, et al. Study of Urban Intersection Lane Structure Based on Floating Car Tracking Data. *Science of Surveying and Mapping*, Vol. 39, No. 12, 2014, pp. 76-80.
10. Zhuang, Lijian, et al. A Large-scale Floating Car Data-based Algorithm of Turning Rule Extraction at Intersections. *Journal of Wuhan University of Technology: Transportation Science & Engineering*, Vol. 37, No. 5, 2013, pp. 1084-1088.
11. Huang, Min. Modeling Theory and Method for Road Guide Sign System. Diss. Guangzhou: Sun Yat-sen University, 2006.
12. Huang, Min, et al. Modeling Method for Road Guide Signs Based on Intersection Features. *Journal of Southwest Jiaotong University*, Vol.42, No. 1, 2007, pp. 110-114.
13. She, Xiwei, et al. Online Map-Matching Framework for Floating-car Data with Low Sampling Rate in Urban Road Network. *Transportation Research Board 91st Annual Meeting*, No. 12-1751, 2012.
14. Zhuang, Lijian, et al. Framework of Experienced Route Planning Based on Taxis' GPS Data. *Intelligent Transportation Systems (ITSC)*, 2012 15th International IEEE Conference on. IEEE, 2012.