J Supercomput (2017) 73:3128–3148
DOI 10.1007/s11227-017-2013-y

Predicting the next turn at road junction from big traffic data

Yan Zhuang[1] · Simon Fong[1] · Meng Yuan[1] · Yunsick Sung[2] · Kyungeun Cho[2] · Raymond K. Wong[3]

Published online: 21 March 2017

Abstract Smart city is an emerging research field nowadays, with emphasis of using big data to enhance citizens' quality of life. One of the prevalent smart city projects is to use big traffic data collected from road users over time, for road planning, traffic light scheduling, traffic jam relief, and public security. In particular, being able to know a road user's current location and predict his/her next move is important in today's intelligent transportation systems. Trajectory prediction has become a prudential research study direction, by which many algorithms have been published before. In this paper, we present a simple probabilistic model which predicts road users' next locations based on the "concept of segments" abstracted from historical trails which the users have taken and accumulated over time in some data archive. Given a trajectory and a current location, the road user's next move in terms of road direction

✉ Simon Fong
ccfong@umac.mo

Yan Zhuang
syz@umac.mo

Meng Yuan
250625774@qq.com

Yunsick Sung
sung@dongguk.edu

Kyungeun Cho
cke@dongguk.edu

Raymond K. Wong
wong@cse.unsw.edu.au

[1] Department of Computer and Information Science, University of Macau, Taipa, Macau SAR

[2] Department of Multimedia Engineering, Dongguk University, Seoul, South Korea

[3] School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

can be predicted at the junction. It is found that each road user would have his/her unique travel pattern hidden in the aggregate big traffic data. These patterns could be modeled from connected segments for simplicity. With the longer the trail and more frequently this trail was travelled, the more accurate that the next turn can be predicted. Simulation experiment was conducted based on summing up the segments from empirical trajectory data that was used in trajectory data mining by Microsoft. The results of our alternative model in contrast to the state of the arts demonstrated good efficacy.

Keywords Location prediction · Trajectory mining · GPS Trajectory analysis

1 Introduction

Over time, we all travel on regular routes, daily, weekly, or monthly that exhibit some specific patterns pertaining to our lifestyles and work schedules. These paths are anything but random, embracing our habitual trails. For instance, students go by regular school buses daily; working professionals shuttle between offices and homes every morning and evening; for recreation purpose, we have our favorite hangout places where we patron regularly. The same idea applies over walking, driving, and riding on public transports that leave our trails behind our activities on the use of the roads. Based on this principle, nowadays many projects of smart cities are much talked and planned about; one of the main parts of smart city synergy [1,2] is about sensing, collecting traffic patterns, analyzing them, and inferring a suitable strategy in traffic/town planning for improving quality of life. Applications that were designed for the sake of enhancing our transportation systems such as crowd control, security surveillance, traffic jam remedies, and better allocation or sharing of road resources have emerged recently. The ability of the underlying decision making, however, hinges on how efficient and effective the analytics is in analyzing and predicting the next moves of every individual.

Positioning technology [3] evolves with increasing popularity recently with better accuracy and affordability. Location-based services for smartphones are so common nowadays [4], which signal the users' current geographic locations to the mobile network and pull information about points of interest from their proximities [5]. Predicting a person's next location where he is moving to, from his current position is becoming mandatorily important for location-based analytics which in turn supports intelligent transportation systems [6] and location-based services (LBS) [7] for individuals. For an example of unmanned vehicle which is a rising research topic, Nissan admitted that if the route of vehicle could be known and preset in advance, hybrid fuel economy can be saved by 7.8% [8]. It is seconded by Tate and Boyd who invented some optimal control scheme for a hybrid vehicle assuming the route is already known [9].

For predicting a traveler's trajectory given his current position and his historical path, a number of algorithms exist [9–12]. These algorithms usually attempt to match a predicted future path with the existing path with the lowest fitting error. The algorithms often assume that the trajectory carries elements of people's intentions and destinations, as well as the moving speeds and their changes along the way. They work by connecting the dots along the paths, so to infer the future path for a particular traveler.

In this paper, we investigate an alternative type of trajectory prediction model which simplifies the trajectory details using the concept of segments. The trajectories or paths that have been travelled by all the users are aggregated into travel segments for generalizing them into some representative frequently travelled paths. Each segment is made up of turns at the junctions along the way, thereby ignoring the distance and direction details between the turns. When we piece up a series of segments representing a current trajectory, the next direction of turn given the current junction where the traveler is at, could be predicted according to the segments which he has travelled through. This "next-turn" prediction model works by computing through the history of all the travelers' paths as to match the current traveler's path and his next turn. In other words, our new model does not require detailed matching. The next turn of a path which we are interested to know is inferred from a similar representative travel pattern summed from an aggregated archive of data. There is hence no need to collect sufficient path records for an individual traveler nor digging into details of each path travelled by the individual traveler. It is therefore more suitable for big data analytics [13] where often mass data (instead of individual profile) are dealt with. Efficiency is anticipated by our model because we simply match the segments which are already generated and transformed from the original raw data in the preprocessing step upon adding in new records from the sensors to the database. The segment database is believed to well represent the purposes and traveling patterns of the travelers without explicitly recording the details. For example, a high traffic flow of high traveling probability from the start to the end of a highway connecting some important suburbs to the central business districts in the weekday mornings, represents a popular route for people arriving office for work. Their turns in terms of entries and exists of the highways could be represented by probabilities for guessing where the mass would go. Other examples include but not limited to a frequently travelled route linking from public transport stations to tourist attractions, occasional heavy packs of vehicle flow to and from stadiums during events, popular transits at certain intersections connecting people living and working in different areas, and several routes that direct people to and from an airport, etc.

We predict by matching the current trajectory without regard to any particular traveler's historical paths. Using concept of travelled segments that facilitates rough representation of trails accumulated, and collective segments that are summed over from the population in the archive, help simplify the computation significantly.

The remaining of this paper is structured as follows. Section 2 presents the related works about some possible application scenarios as well as the similar trajectory prediction techniques. In Sect. 3, the prediction model is described. Simulation experiments and the results are reported in Sect. 4. Section 5 concludes the paper.

2 Related work

2.1 Application scenarios

Some possible application scenarios are listed below, by using our model to predict the next turn given a current position: (1) Urban security—after a crime took place, the robbers were on the run. With the aid of real-time CCTV cameras that are streaming in live data feed [14], the policemen know about the trajectory of robbers have so far taken. Without knowing their identities and being unable to know their destinations, the robbers were last seen at a road junction. In this situation, the police would want to know the probabilities of the subsequent turns from the junction, given the path which the robbers travelled from so far, so to continue the pursuit. Our model predicts the most probable turn (or direction) that a traveler would be most likely taking provided that the path pattern that they came from. (2) LBS recommendation service [15]—we assume the traveler who is a tourist has visited a series of spots of interest. From the path that connected a number of tourist attractions that he traveled, we could recommend the next destination this tourist should visit. The recommendation is based on the most probable next turn given his current location, as most people have already travelled in such style. (3) Road navigation system for vehicle drivers—for vehicle drivers, since their trajectories have been tracked as a series of segments, at each turn, probabilities of traffic flows could be displayed to the drivers. This additional information offers drivers information of which turns of the road from this junction that most vehicle drivers have mostly taken given the routes that they have traveled. This is useful when explicit destination is not known nor stored in the system, drivers drive by go-with-the-flow strategy that follows the most probable route which has been taken by drivers in the past who travelled the same routes to the current junction. (4) Road planning—by knowing the probabilities of which turns at each junction, simulation model could be better conducted in modeling the road traffic usage, and give better results especially in what-if scenarios planning. (5) Catching the catch—the same idea by this model could be applied in lieu of human on animals that move, for example, animal hunting, netting fish, and intercepting moving targets on the road. At different time periods and certain season, fish in the wild usually swim pass a specific place, from one stream to another. When their historical paths are known, fishermen could predict by when and where they will be making a bigger catch. The same principle applies to anything that is on the run over the road network [16]; knowing where they are from, where they are now, the model predicts their next turns from the collective big data.

2.2 Trajectory inference models

Pu [10] gave a method based on dead-reckoning. She predicted the next latitude and longitude by the current latitude, longitude, bearing, in consideration of speed and time intervals. According to her method, when these elements are obtained, multiply speed by travel time is the traveled distance. The next latitude and longitude could be predicted by current latitude, longitude, bearing, and distance. Using this algorithm, a remote system could get to know the real-time position of the pedestrian. It was argued that GPS signal could be affected by interference via various factors such as weather, obstacles of buildings, blind spots of electromagnet fields in underground and enclosed lifts, etc. Even if GPS signal could be received normally, there is also delay on remote monitor. Therefore, by using complex ingredients in predicting a result, the result at best is an approximate from the actual situation.

In Liu and Karimi's [17] algorithm, they consider the current location of the person to be predicted as the center of a circle, in considering the speed limits on the roads that multiplies the prediction period as the radius of the circle. Exit points are denoted as the points at the circle circumference that intersect with the road. A graph search tree is constructed by taking current location as the root, and exit points as leaf nodes. From the root to a leaf, the shortest paths are regarded as the possible paths. There is a probability for taking each node. According to the probabilistic model, from current position to each exit point, the probabilities could be calculated. The path that has the highest probability is the most possible route. Coverage of a radius of circle is considered in this method, which is unlike ours. We instead centered on the road junctions by approximating the segment lengths instead of precise distance from circle radius.

In Ye's [18] algorithm, they mined the pedestrian's route to generate route of interest (ROI). Then all the ROI points are integrated and transformed into a prediction tree. Based on the tree, a prediction is made to predict the next ROI of the pedestrian. In this method, similar concept of simplifying the detailed routes like ours was used. We took it to a higher level of abstraction by summing up the routes traveled from all the travelers.

Anagnostopoulos [19] proposed a context model based on classification, which deals with location prediction of moving users. The induced model predicts the next movement of a mobile user with certain moving profile and history of movements. Then temporal context periods throughout a day are also incorporated in the proposed model. Two classification schemes are introduced in order to support location prediction. These schemes are evaluated with three data mining algorithms, and the most accurate algorithm is adopted. Finally, the classification schemes are also compared with three non-data mining schemes for location prediction, by means of prediction accuracy.

In Liao's [20] method, they introduced a hierarchical Markov model to learn and infer a user's daily movement through an urban community. Multiple levels of abstraction are used to bridge the gap between raw GPS data and high-level information such as user's destination and mode of transportation. Rao-Blackwell utilized particle filters to achieve efficient inference. Locations where the user frequently changes mode of transportation, such as bus stops and parking lots, are learned from GPS data logs without manual labeling of training data. Then it detects the novel behavior or user errors by modeling activities in the context of the user's historical data. Recently, there is an emerging research trend with analytics applied on big road traffic data. They range from using big data to predict highway traffic accident [21], analysis of transportation system [22], to vehicle speed estimation [23], etc.

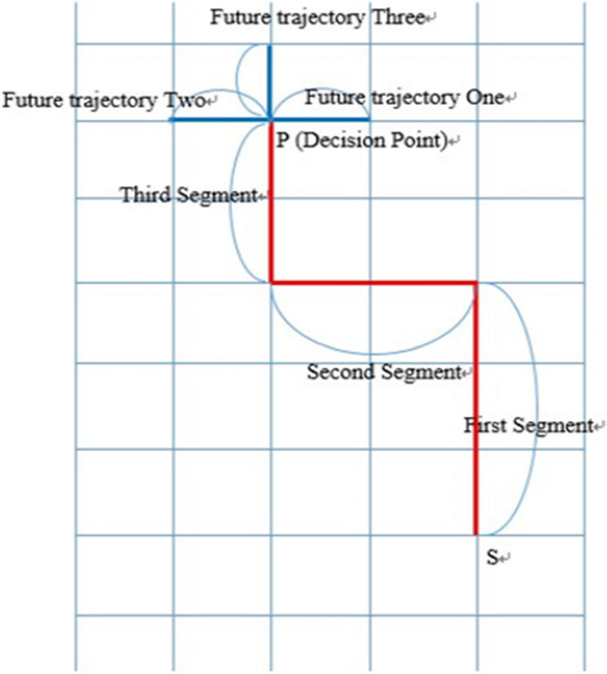


Fig. 1 Illustration of segment, current trajectory, and future trajectory

3 Segment-based prediction model

3.1 Model formulation

The concept of segments, current trajectory, and future trajectory is shown in Fig. 1. The objective is to predict which turn from a junction where a traveler will continue from his current position. The junction is called decision point. The travelled paths (from markers S to P as shown in Fig. 1) are characterized by segments which is a series of junctions connected in some order.

The formulation of the prediction model is based on the layout of a road map where the accessible routes and junctions are shown. In the simplest form, the entire map contains information, $M = <V, E>$. The junctions in the map are represented by a collection of vertices, V. A vertex in M is denoted by a group of $v_i(i \in |N|)$. $|N|$ is the number of possible branches spanning out from the junction. A series of junctions which a traveler has passed is an ordered vector denoted by $\rho_j = (v_1, v_2, v_3, \ldots v_M)$. The current junction where the position of the traveler is currently at is called a decision point v_D. The next junction (or direction of the turn) that the traveler is predicted to turn into from v_D is v_N, which is coined as the future vertex. $s_D \in E$ is a series of vertices that the traveler has passed through before reaching v_D. It is called current trajectory that has a list of visiting records. At the decision point, v_D, the probability of taking a future turn v_N is shown in Eq. 1, $P(\bullet)$.

$$\begin{aligned} P\left(v_N, v_1, v_2, v_3, \ldots v_D\right) &= P\left(v_N | s_D\right) \\ P\left(v_N, s_D\right) &= P\left(v_N | s_D\right) \end{aligned} \tag{1}$$

where $v_1, v_2, v_3, \ldots v_D$ are transformed into a vector of segments as s_D. The segment vector is a simplified version of route that the travelers have already passed through $v_1, v_2, v_3, \ldots v_{D-1}$, prior to arriving at the decision point v_D. $P(v_1, v_2, v_3, \ldots v_D)$ is the summation of the probabilities in terms of frequency counting of taking $v_1, v_2, v_3, \ldots v_D$.

$P(v_N|s_D)$ defines the probability of taking v_N when the traveler walked through $v_1, v_2, v_3, \ldots v_{D-1}$ that leads to the decision point v_D. When the traveler is now standing at the crossroad at v_D, which is the decision point, the probability matrix of taking each future vertex is shown as follows:

$$\overline{P(M)} = \begin{bmatrix} P\left(v_1|s_k, s_1\right), & P\left(v_2|s_k, s_1\right) & \cdots & P\left(v_i|s_k, s_1\right) \\ \vdots & & \ddots & \vdots \\ P\left(v_1|s_k, s_{j-1}\right), & P\left(v_2|s_k, s_{j-1}\right) & \cdots & P\left(v_i|s_k, s_{j-1}\right) \\ P\left(v_1|s_k, s_j\right), & P\left(v_2|s_k, s_j\right) & & P\left(v_i|s_k, s_j\right) \end{bmatrix} \forall\, i, j, k \tag{2}$$

In this prediction matrix, conditional probability $P\left(v_i|s_k, s_j\right)$ is the probability of the traveler who has passed s_{D-1} finally getting to v_i along the trajectory s_j. v$_i$ is future vertex like v_N. $s_k$is the trajectory the traveler has passed. s_j is a series of vertices the traveler has taken from v_D to v_i. Sum of a column in Eq. 2 could be expressed as follows:

$$\begin{aligned} \overline{P(M)} &= \left[\sum_{z=1}^{j} P\left(v_1|s_k, s_z\right), \sum_{z=1}^{j} P\left(v_2|s_k, s_z\right), \ldots \sum_{z=1}^{j} P\left(v_i|s_k, s_z\right)\right] \\ &= \left[\frac{|s_1, v_1|}{|s_k|}, \frac{|s_1, v_2|}{|s_k|}, \ldots \frac{|s_1, v_j|}{|s_k|}\right] \end{aligned} \tag{3}$$

where s_k is the trajectory of the traveler who has passed the sequence of vertices $v_1, v_2, v_3 \ldots v_D$. $|s_k|$ is the historical number of travelers who have passed this path crossing through the vertices sequence $v_1, v_2, v_3 \ldots v_D$ in the path history database. $|s_k, v_N|$ is the number of times that travelers have passed the sequence $v_1, v_2, v_3 \ldots v_D$ as historical path, leading to the next vertex v_N.

$\sum_{z=1}^{j} P\left(v_i|s_k, s_z\right)$ is the probability of a traveler who has passed s_k segments, finally reaching the point v_i. Between the decision points v_D and v_i, the traveler has passed s_k.

From Eq. (3), the probability of a traveler who has passed a series of vertices represented by s_k and eventually arrives at one of the possible candidate future vertex v'_N is shown as follows:

$$P\left(v'_N, s_k\right) = \sum_{z=1}^{j} P\left(v'_N|s_k, s_z\right) = \frac{|s_k, v'_N|}{|s_k|} \tag{4}$$

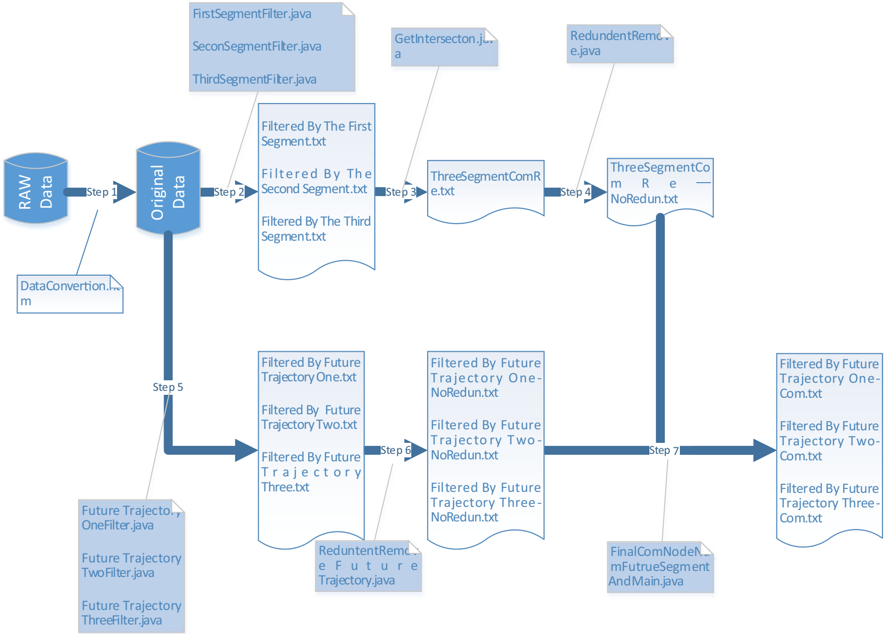

FirstSegmentFilter.java
SeconSegmentFilter.java
ThirdSegmentFilter.java
GetIntersecton.java
RedundentRemove.java
RAW Data
Step 1
Original Data
Step 2
Filtered By The First Segment.txt
Filtered By The Second Segment.txt
Filtered By The Third Segment.txt
Step 3
ThreeSegmentComRe.txt
Step 4
ThreeSegmentCom Re — NoRedun.txt
DataConvertion.m
Step 5
Filtered By Future Trajectory One.txt
Filtered By Future Trajectory Two.txt
Filtered By Future Trajectory Three.txt
Step 6
Filtered By Future Trajectory One-NoRedun.txt
Filtered By Future Trajectory Two-NoRedun.txt
Filtered By Future Trajectory Three-NoRedun.txt
Step 7
Filtered By Future Trajectory One-Com.txt
Filtered By Future Trajectory Two-Com.txt
Filtered By Future Trajectory Three-Com.txt
Future Trajectory OneFilter.java
Future Trajectory TwoFilter.java
Future Trajectory ThreeFilter.java
RedunteentRemove Future Trajectory.java
FinalComNodeNumFutrueSegment AndMain.java

Fig. 2 Workflow of the general model setup procedures

The most probable next turn is calculated as maximum $(P(v_N, s_k))$.

3.2 Prediction procedures

One prime ingredient for the prediction is the raw sensing data collected over time, marking the locations with time-stamps traversed by all the road users. In a general sense, the road users can be any cars, bikes, or pedestrians. In our model, a minimum of three segments are used to mimic the shape of current trajectory, which would be used to match it against the historical paths. Figure 2 shows a step-by-step workflow that preprocesses the raw data to segments in terms of path IDs for establishing a trajectory. The raw data format consists of data fields such as latitude, longitude, altitude, and a time-stamp in the form of day of the week, date, and time. The converted data format has the same with an automatically generated path ID number appended; the time-stamp is simplified too for fast processing.

A real-life GPS dataset from Microsoft Research Asia[1] which contains 17,687 trajectories is used in the experimentation for validating the proposed model. This dataset recoded a broad range of users' outdoor movements, including not only life routines like going home and going to work but also some entertainments and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. It is widely

[1] https://www.microsoft.com/en-us/research/publication/gps-trajectories-with-transportation-mode-labels/.

distributed over 30 cities of China and even in some cities located in the USA and Europe. The majority of the data was created in Beijing, China. Every single folder of this dataset stores a user's GPS log files in PLT format. PLT files are vector image files primarily associated with a graphic design program called AutoCAD. Each PLT file contains a single trajectory and is named by its starting time. So, there are 17,687 PLT files in total. The files contain header fields such as information of latitude, longitude, height, speed, and heading direction. After preprocessing, we retained information of latitude, longitude, and time-stamp as temporal–spatial data.

4 Experiments

The experiments are designed with the objective of verifying the possibility of the proposed segment-based model. Pertaining to the design of the prediction model, there are four main aspects that may vary in the model which in turn might affect the performance of the model; they are 1) different segment numbers that compose a trajectory; 2) different trajectory or path lengths that are already known; 3) different shapes of the trajectory; and 4) effects of time filters. The dataset contains path information of both past and future. By referring to the time-stamps, we can control using the paths that are simulated to be those that have been travelled, and those so-called future paths. Thus, it is possible to validate the accuracy of the prediction model since the subsequent future paths are already known in advance as they were stored in a database in the testing, which have extended out from the travelled paths.

4.1 Testing with different numbers of segments

By default, we just arbitrarily choose three random segments for fitting a trajectory that contains information of locations that have been travelled. Usually any number of segments can be used to form a trajectory. If a different number of segments are used to match the shape of the current trajectory, we may probably get different results. So, in this part of the experiment we test different numbers of segments and check the effect on the prediction accuracy. We randomly select some testing paths that are made up of 2 to 5 segments. Paths of path IDs, #2, #3, #5, and #9 have been selected (the ID numbers are just arbitrary). It was found that a path that is made up of five segment has covered most of the complex shapes and that has almost reached the maximum length of paths that most people would have travelled in a city. Five segments are sufficient to cover even the most complex travel patterns without any recurrence, or leading back to their original places, e.g., return trips. The paths selected here are those that took place within half a day (12 hours). They all have unique starting positions and ending positions, as one-way trips. The results are given in Table 1 which shows the probabilities of the possible turns. The turns that have satisfactory good probabilities are chosen as the predicted turn. They are marked as italics in the table. Then, the predicted turn is checked against the actual turn from the data archive. The same experiment is repeated 20 times with the same setting but with different randomly picked paths from the large data archive. The average accuracies are charted in Fig. 3.

Table 1 Experiment results of different numbers of segments used

Path ID (#Segment)	Probability (%)			
	Future trajectory 1	Future trajectory 2	Future trajectory 3	Most probable future trajectory
2 (2)	*80.71*	8.15	0.71	80.71
2 (3)	*83.58*	15.76	0.75	83.58
2 (4)	*83.61*	37.99	0.82	83.61
2 (5)	*83.61*	37.99	0.82	83.61
3 (2)	5.04	*85.71*	9.24	85.71
3 (3)	3.16	*88.42*	8.42	88.42
3 (4)	3.37	*89.89*	6.74	89.89
3 (5)	3.37	*89.89*	6.74	89.89
5 (2)	9.43	*89.62*	0.94	89.62
5 (3)	9.61	*89.42*	0.96	89.42
5 (4)	9.61	*89.42*	0.96	89.42
9 (2)	*98.65*	0	1.35	98.65
9 (3)	*98.65*	0	1.35	98.65
9 (4)	*98.65*	0	1.35	98.65

Italic values indicate the best performance results

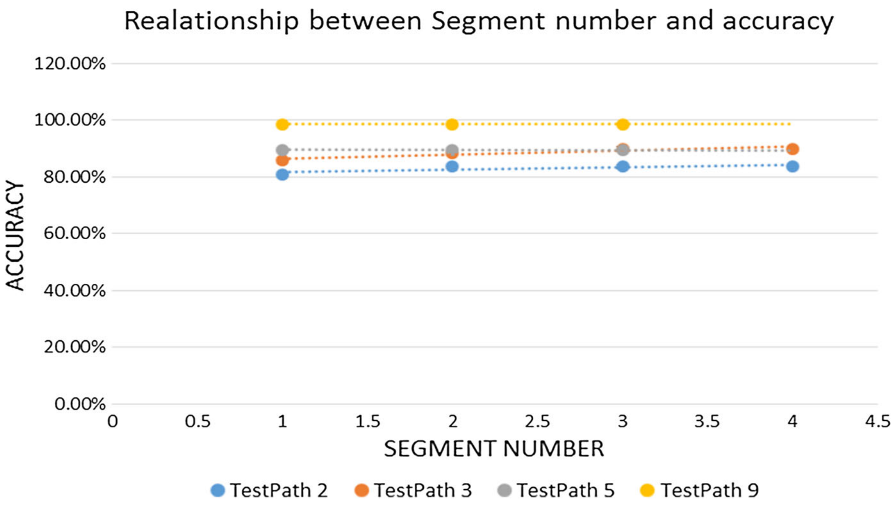


Fig. 3 Average results of different numbers of segments used

Figure 3 shows that each line stays almost flat when the number of used segments varies. Again, in Table 1, we could observe that the number of segments affects just slightly on the accuracy. For this reason, we compute the accuracy changing rate based on the segment variety using a simple formula $\frac{\Delta_{accuracy}}{\Delta_{segment\ number}}$. The accuracy changing rates for the four randomly chosen paths, after averaged, are: 0.966667,

Table 2 Experiment results of different lengths of trajectory used

Path ID. (distance)	Probability (%)			
	Future trajectory 1	Future trajectory 2	Future trajectory 3	Most probable future trajectory
3 (890 m)	5.73	*84.07*	10.19	84.07
3 (1217 m)	5.51	*85.52*	8.97	85.52
3 (1577 m)	3.16	*88.42*	8.42	88.42
3 (2193 m)	1.28	*91.03*	7.69	91.03
6 (759 m)	21.85	21.85	*56.29*	56.29
6 (1259 m)	20.27	18.88	*60.84*	60.84
6 (1762 m)	19.51	18.70	*61.79*	61.79
9 (1613 m)	*96.05*	1.97	1.97	96.05
9 (2469 m)	*97.46*	1.69	0.87	97.46
9 (3329 m)	*98.65*	0	1.35	98.65
9 (3756 m)	*98.61*	0	1.39	98.61
10 (1427 m)	16.67	19.61	*63.73*	63.73
10 (2773 m)	23.61	12.50	*63.89*	63.89
10 (4157 m)	7.89	10.52	*81.58*	81.58

Italic values indicate the best performance results

1.393333, −0.1, and 0.00054%, respectively. Averaging out the changing rates, we have 0.565%. As a result, the accuracy only changes about 0.565% upon increasing or decreasing one segment in average. The effect of segment amounts is quite insignificant.

4.2 Testing with different lengths of trajectory

In this part, we test current trajectory with different current trajectory lengths in order to investigate the relationship between current trajectory lengths and accuracy. In this test group, 2 to 4 segments are used to fit the shape of the current trajectory. The same experiment setting is kept; 20 sample runs are drawn randomly from the data archive. The results are given in Table 2. Again those paths that produce sufficiently high probabilities are marked in shaded cells. The average accuracies from the 20 samples are averaged and charted in Fig. 4.

Figure 4 shows that the average accuracy of guessing the future turn correctly from each test path rises as the current trajectory length increases. Cross-checking with the results in Table 2, we can see that the accuracy improves by one degree on every 100 meters of extra distance added. From the above data, we confirm that the correct prediction accuracy is in a direct ratio to the current trajectory length. When the current trajectory increases per 100 meters, the accuracy will grow 0.46418625% in average.

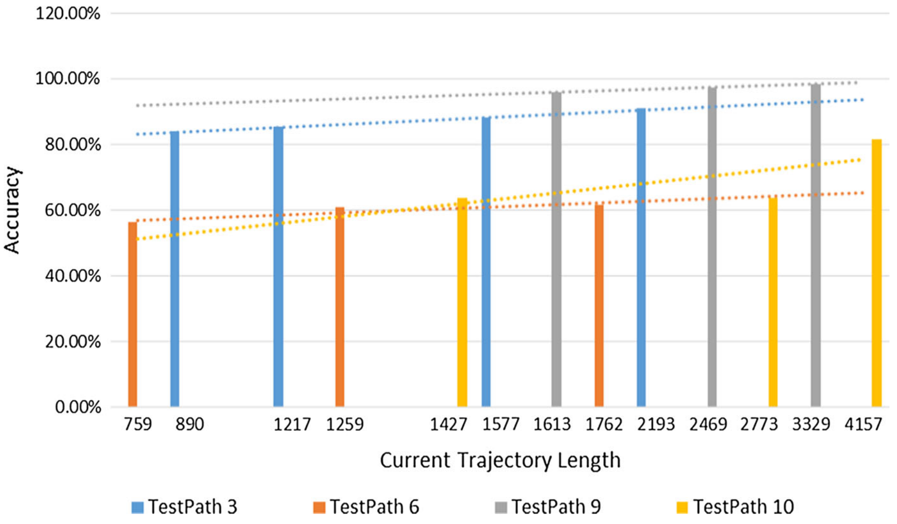


Fig. 4 Average results of different lengths of paths used

4.3 Testing with different complexities of trajectory

Here, in this part of the experiment, we pick ten test paths randomly and place them into two groups according to their trajectory characters. In general, group A is more straightforward than group B. The trajectories in different groups have different twists and turns. The objective of this experiment is to evaluate the performance of the prediction model under different shapes of trajectory. The same experiment setting is used, and 20 samples from each path are taken for evaluation. The results are averaged out and charted in Figs. 5 and 6 for group A and group B, respectively. The selected ten paths are given in "Appendix."

From Figs. 5 and 6 that indicate the performance by groups A and B, respectively, the accuracy variations for each test in groups A and B are estimated by the following formula. Average accuracy for test $i = \frac{\sum_i Accuracy(i)}{\sum_i Length(i)}$. So,

Group A: Average accuracy for group A = $\frac{447.77\%}{14479} = 0.030925\%$
Group B: Average accuracy for group B = $\frac{359.42\%}{7418} = 0.04845241\%$

According to the group results, it is reasoned that as the current trajectory length increases, the collective accuracy of the paths in group B increases more than that of the paths in group A. That means for the same level of path length, higher prediction accuracy can be achieved for the paths with more complex characters than those with simple structure. In other words, complex structures help better characterize the paths and hence they can identify the traffic flow more strongly and uniquely.

To sum up thus far, we could get the following conclusion by observing the experiment results.

1. Accuracy is generally improved when the current trajectory length increases.

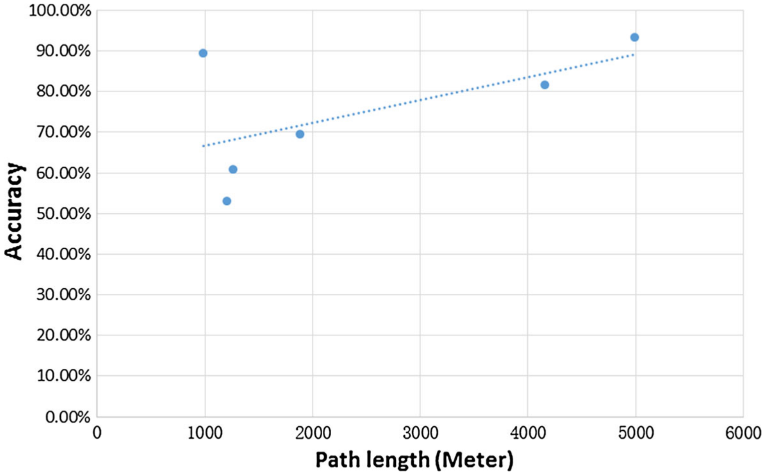


Fig. 5 Average accuracy versus path length for group A of paths

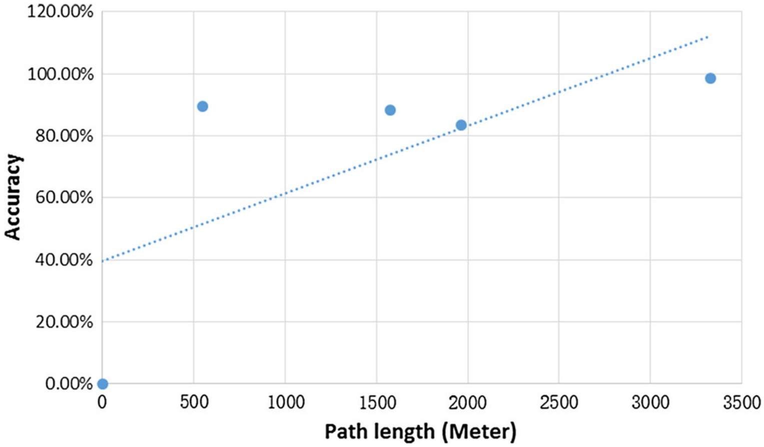


Fig. 6 Average accuracy versus path length for group B of paths

2. For different character level, in the same trajectory length level, we could get a higher accuracy from the group with complex trajectory character which is characterized by more twists and turns.

So, the prediction accuracy is dependent on the current trajectory length and trajectory character in terms of different turns.

4.4 Testing with time filter

In reality, traveled traits are time related which are due to the fact that people (travelers) do travel by some time schedules. For example, certain roads are more traveled than

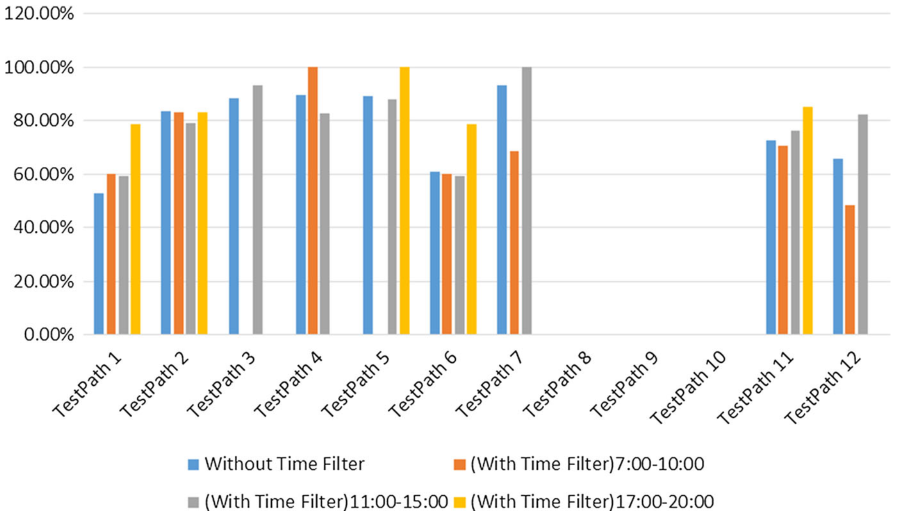


Fig. 7 Average accuracy of prediction with and without time filters

the others during business peak hours, schooling times, and religious worships that go by certain hours of the day and days of the week.

Within certain time periods, a large number of people do the same thing in common. In the morning most of us do go work/study, till afternoon or evening we return home. At noon, people seek for lunch. Therefore, our prediction model can be modified with a time filter that caters for refining the predictions in account of different time periods of a day. In this modification, we match current trajectory with historical path which was gathered from within a specified time period instead of all the historical paths from the database.

Let δ be the time period of a day.

Let $\delta(|s_k|)$ be the number of historical paths that travelers have traversed through the vertices sequence $v_1, v_2, v_3 \ldots v_D$ during the time period δ.

$\delta(|s_k, v_k|)$ is the number of historical paths that travelers have traversed through the vertices sequence $v_1, v_2, v_3 \ldots v_D$ and subsequently reached v_k, during the time period δ. Substituting into Eq. 2, the time-based prediction could be approximated by $\frac{\delta(|\mathrm{S}_k, v'_N|)}{\delta(|s_k|)}$.

In our experiment, three main time periods are used, and they are 7:00 to 10:00, 11:00 to 15:00, and 17:00 to 20:00 which generally represent morning, noon, and evening. By imposing the time periods over the 12 paths, with 20 samples for each path, the results are simulated and given in the following table. The average accuracies are charted in Fig. 7.

The first column is just a list of arbitrary IDs of test paths; the second column lists the results without time period constraint. And we ignored the results with little data support such as test path 8, test path 9, and test path 10, because there are few records available during those times. Hence, the probability in this situation drops to almost zero (≈0%). From the results given in Table 3, we compare the results without

Table 3 Experiment results when time filter is applied

	Result	(7:00–10:00)				(11:00–15:00)				(17:00–20:00)			
		CT	FT1	FT2	FT3	CT	FT1	FT2	FT3	CT	FT1	FT2	FT3
1	*53.03% (B3)*	39	2 (3.64%)	20 (36.36%)	*33 (60%)*	63	0 (0%)	*42 (59.15%)*	29 (40.85%)	30	1 (3.57%)	5 (17.86%)	*22 (78.57%)*
2	*83.58% (B1)*	15	*15 (83.33%)*	3 (16.67%)	0 (0%)	18	*15 (78.95%)*	4 (21.05%)	0 (0%)	63	*50 (83.33%)*	10 (16.67%)	0 (0%)
3	*88.42% (B2)*	**3**	**0 (0%)**	**2 (100%)**	**0 (0%)**	61	1 (1.69%)	*55 (93.22%)*	3 (5.08%)	3	0 (0%)	*5 (100%)*	0 (0%)
4	*89.47% (B3)*	27	0 (0%)	0 (0%)	*22 (100%)*	43	8 (17.02%)	0 (0%)	*39 (82.98%)*	0	0 (0%)	0 (0%)	0 (0%)
5	*89.42% (B2)*	**8**	**2 (50%)**	**1 (25%)**	1 (25%)	26	2 (11.76%)	*15 (88.23%)*	0 (0%)	14	0 (0%)	*12 (100%)*	0 (0%)
6	*60.84% (B3)*	39	2 (3.64%)	20 (36.36%)	*33 (60%)*	63	0 (0%)	*42 (59.15%)*	29 (40.85%)	30	1 (3.57%)	5 (17.86%)	*22 (78.57%)*
7	*93.27% (B2)*	16	4 (15.38%)	*11 (68.75%)*	1 (6.25%)	47	0 (0%)	*48 (100%)*	0 (0%)	**1**	**0 (0%)**	**1 (100%)**	**0 (0%)**
8	*69.57% (B1)*	**8**	**11 (91.67%)**	**0 (0%)**	**1 (8.33%)**	**9**	**5 (55.56%)**	**1 (11.11%)**	**3 (33.33%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**
9	*98.65% (B1)*	**7**	7 **(87.5%)**	**0 (0%)**	**1 (12.5%)**	**6**	**6 (100%)**	**0 (0%)**	**0 (0%)**	**0**	**0 (0%)**	**0 (0%)**	**0 (0%)**
10	*81.58% (B3)*	**2**	**0 (0%)**	**0 (0%)**	**1 (100%)**	**3**	**0 (0%)**	**0 (0%)**	**3 (100%)**	**2**	**0 (0%)**	**0 (0%)**	**2 (100%)**
11	*72.79% (B1)*	106	*79 (70.54%)*	17 (15.18%)	16 (14.29%)	246	*179 (76.5%)*	27 (11.54%)	28 (11.97%)	29	*23 (85.19%)*	1 (3.7%)	3 (11.11%)
12	*65.65% (B3)*	30	3 (9.09%)	*16 (48.49%)*	14 (42.42%)	148	1 (0.77%)	22 (16.92%)	*107 (82.31%)*	**1**	**0 (0%)**	**0 (0%)**	**1 (100%)**

Italic values indicate the best performance results
CT current trajectory
FT future trajectory

time period constraint with the results with time period constraint. The prediction accuracies are averaged over the samples and charted in Fig. 7.

As shown in Fig. 7, we could conclude that generally for each test path whenever time filter is applied, there is an improvement in prediction accuracy result compared to the one without time filter, in most cases. Some time filters show strong difference in results such as the morning period and evening period. The evening period is strongest in influencing the prediction. It is supposed that there would be very high volume of traffic (even traffic jams) that occurs during those peak hours, where almost everybody is coming home out of the city by taking those limited routes. The experiment results do show that time filters have influence on the prediction results and accuracies. For instance, in test path 6, the result without time constraint is predicted at future trajectory 3 with 60.84% probability, but the result of time period from 11:00 to 15:00 is at future trajectory 2 with 59.15% probability. If the predicted trajectory took place in the period of 11:00 to 15:00, it would rather be future trajectory 2 with 59.15%.

Comparing our prediction model with some classical states of the arts like Liu [17], Ye [18], Liao [11], and Jeung's [12] algorithms, we offer an alternative model which does not cater for predicting the next exact locations. Rather, our model approximates about which turn out of all possible turns from a road junction a traveler is likely to take after he has been traveling certain length of path. This approximate prediction model highly simplifies the data volume required for big transport data, because information regarding the traveling velocities and profiling of each individual road user are not required. It is based on the belief that the traveling patterns for each individual user would not fluctuate too much over time. So collectively when the traveling patterns for all the people in a city have been aggregated the next-turn probabilities at each road junction would become quite stable, which is resilient to some abrupt changes by some minority. If ever the probabilities do get change, it implies there are some changes in trends of traffic movements, which may trigger the need for road traffic re-planning.

In terms of data processing speed, which depends on the underlying database structure and the search algorithm being used, typically the probability inference takes about few to dozens of minutes. Most of the time was spent in scanning the database, but the probability computation is extremely simple and fast. Once the probabilities are computed and stored in place for each road junction, the testing is very fast without the need of referring to the data archive. Hence, this model is suitable for big data analytics for quick referencing the likelihood of taking which turn at a junction, and guessing where the traveling object [20] would proceed next, given the trajectory he has traveled so far.

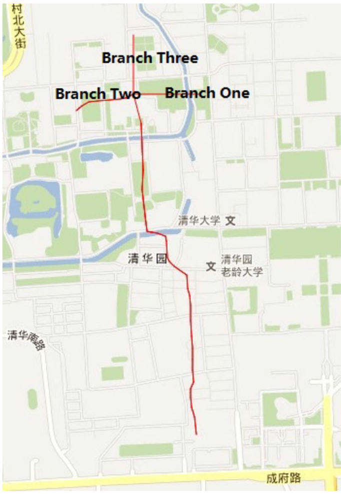
5 Conclusion

In recent years, with the rapid popularization of the vehicle navigation equipment, intelligent mobile phones and portable computing devices, the collection of the trajectory data has become more convenient than ever. The historical trajectory data of people or cars contain important information about their movements, and through analyzing it, we can predict the user's next location, which can provide more variety of services for the users. The factors affecting the location prediction differ like path distance, weather condition, and landform analysis. In this paper, a novel next-turn prediction model is formulated and studied. The next-turn prediction model predicts the most likely turn that a traveler would proceed from a road junction after he has travelled through certain pattern on the roads. Different from other similar prediction models, our model is inferred from collective traveling data from the mass. Matching precisely on individual users' past records is not needed; therefore, user profiling is not required. By using mass data, the prediction can be refined by time filter which matches from path histories from only certain time periods instead of all. Our experiment results show that there is some influence on the length of past trajectory that a user has traveled to the accuracy of next-turn prediction. The complexity of the trajectory has certain effects on the accuracy too, because trajectory which takes a complex (hence relatively unique shape) can better identify and predict the future turns at road junctions ahead.

As future work, more experiments would be devoted in modeling this phenomenon of trajectory complexity in relation to next-turn prediction accuracy. It is interesting also to associate the unique patterns with different roles or purposes of travels, should sufficient information would become available in the future. This will give more dimensions of information about how the roads were used, the predicted future road uses, and the directional flows of traffic that associate with purposes of uses. Furthermore, the prediction model is anticipated to be enhanced by knowing only which particular turns that a traveler has made, we would be able to predict his next turn at the forth junction. Without referencing to the past trajectories, the prediction can be made more quickly enabling it suitable for real-time prediction that can be made at the nick of time. This work can potentially be extended to optimal route planning or traffic engineering because insights are harvested in terms of probabilities of next turn at any junction given the big traffic data.

Acknowledgements The authors of this paper are thankful to the financial supports of the grant offered with code: MYRG2015-00024, called "Building Sustainable Knowledge Networks through Online Communities," by RDAO, University of Macau. We acknowledge and thank you for the intellectual and technical contributions from Mr. Jia Cong Mao who is a software engineer of DACC laboratory and MSc graduate from the University of Macau.

Appendix

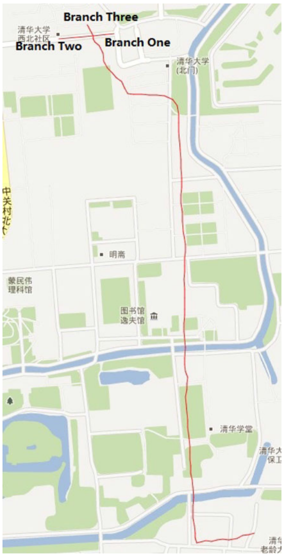


TestPath 1

Length: 1203 m
Probability: 53.03%

Group: **A**

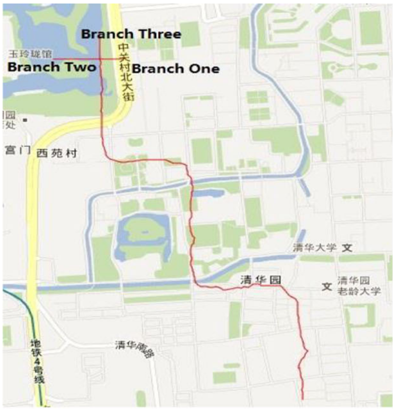
Analysis**:** Though the path measures 1203 meters, it is almost straight.



TestPath 2

Length: 1966 m
Probability: 81.27%
Group: **B**

Analysis: It is 1966 meters long, and it has two turns which is almost 90 degrees.



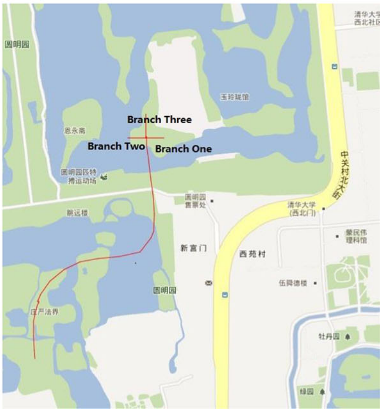
TestPath 3

Length: 1577 m
Probability: 88.42%

Group: **B**

Analysis: It contains four 90 degrees turns.

概率高原因：路线特征明显--弯多，路线长

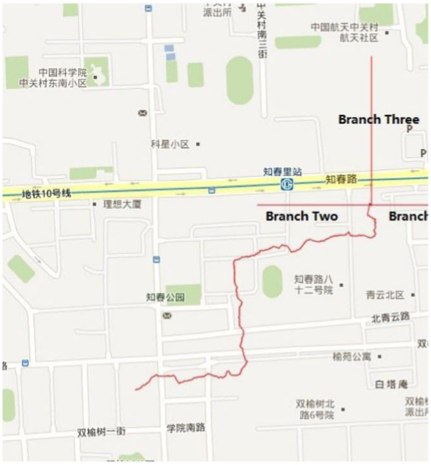


TestPath 4

Length: 984 m
Probability: 89.48%

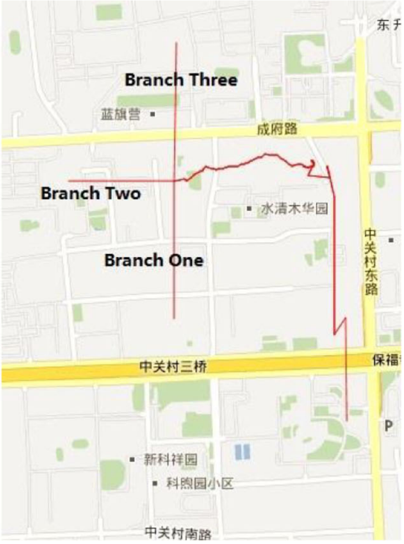
Group: **A**

Analysis: There are two turns at almost 90 degrees. But the length is not long enough and this test path is on the main road.

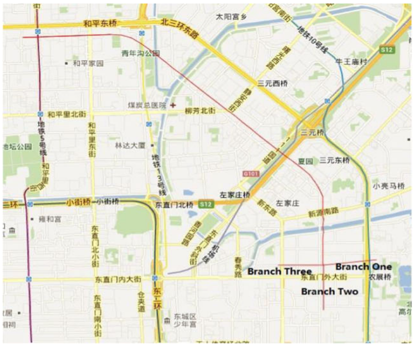


TestPath 5
Length: 546 m
Probability: 91.08%

Group: **B**
Analysis: There are three turns in the trajectory.

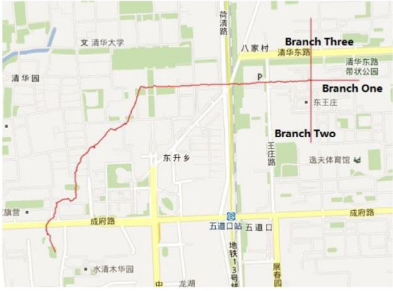


TestPath 6
Length: 1259 m
Probability: 60.84%
Group: **A**
Analysis: There is only one turn in the trajectory.



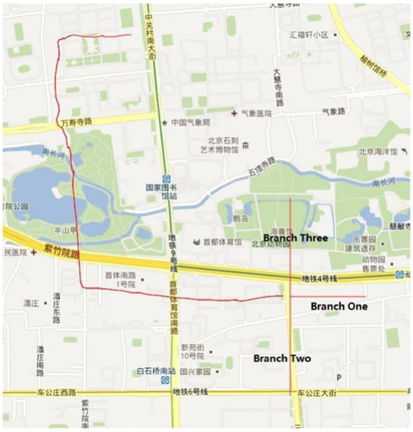
TestPath 7
Length: 4991 m
Probability: 93.27%

Group: **B**
Analysis: It contains two turns at a degree larger than 90.



TestPath 8
Length: 1885m
Probability: 69.57%

Group: **A**
Analysis: It only contains one turn at a degree larger than 90.



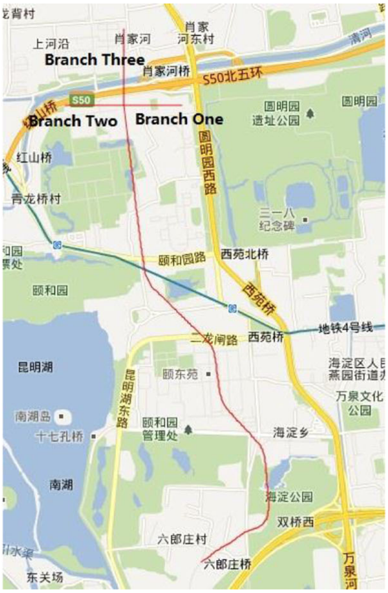
TestPath 9

Length: 3329m

Probability: 98.65%

Group: **B**

Analysis: There are two turns at 90 degree. More importantly, the path takes a U shape.



TestPath 10

Length: 4157 m

Probability: 81.58%

Group: **A**

Analysis: There only one turn in the beginning. And it is at the main road.

References

1. Manaka A, Ishii H, Chow CO, Utsu K (2016) Concepts, services, and underlying network for a daily life support system for citizens. J Supercomput 72(4):1381–1398
2. Jaehak Y, Bang H-C, Lee H, Lee YS (2016) Adaptive internet of things and web of things convergence platform for internet of reality services. J Supercomput 72(1):84–102
3. Lee C, Han K, Li J (2016) Editorial: a special section on ”Emerging platform technologies”. J Supercomput 72(1):1–4
4. Feese S, Burscher MJ, Jonas K, Tröster G (2014) Sensing spatial and temporal coordination in teams using the smartphone. Hum Cent Comput Inf Sci 4(15):1–18
5. Smyth S (2000) Mobile geographic information services: turning GIS inside out. Microsoft Report
6. Joseph AD (2006) Intelligent transportation systems. IEEE J Pervasive Comput 5(4):63–67
7. Dey A, Hightower J, de Nigel LED (2010) Location-based services. IEEE J Pervasive Comput 9(1):11–12
8. Deguchi Y, Kuroda K, Shouji M, Kawabe T (2004) HEV charge/discharge control system based on car navigation information. SAE Convergence International Congress & Exposition on Transportation Electronics, 21-0028, pp 1–7
9. Tate ED, Boyd SP (2000) Finding ultimate limits of performance for hybrid electric vehicles. Technical Report, SAE 2000-01-3099
10. Pu Y, Huaqun G, Wong W-C (2010) 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), 30 Nov-2 Dec 2010, pp 135–140
11. Liao a L, Patterson b DJ, Fox a D, Kautz H (2007) Learning and inferring transportation routines. Artif Intell 171(5–6):311–331
12. Jeung H, Yiu ML, Zhou X (2010) Path prediction and predictive range querying in road network databases. VLDB J 19(4):585–602

13. Anagnostopoulos I, Zeadally S, Exposito E (2016) Handling big data: research challenges and future directions. J Supercomput 72(4):1494–1516
14. Cyrus S, Kim SH, Luciano N, Constantinou Giorgos L, Ying CY, Gérard M, Ramakant N, Farnoush B-K (2014) Janus—multi source event detection and collection system for effective surveillance of criminal activity. J Inf Process Syst 10(1):1–22
15. Sung Y, Chul S (2015) Recommender system using periodicity analysis via mining sequential patterns with time-series and FRAT analysis. J Converg 6(1):1–9
16. Loke SW (2016) Heuristics for spatial finding using iterative mobile crowdsourcing. Hum Cent Comput Inf Sci 6(4):1–31
17. Liu X, Karimi HA (2006) Location awareness through trajectory prediction. Comput Environ Urban Syst 30(6):741–756
18. Ye Q, Chen L, Chen G (2008) Predict personal continuous route. 11th International IEEE Conference on Intelligent Transportation Systems, pp 587–592
19. Anagnostopoulos T, Anagnostopoulos CB, Hadjiefthymiades S, Kalousis A, Kyriakakos M (2007) Path prediction through data mining. IEEE International Conference on Pervasive Services, pp 128–135
20. Lee J, Chae H, Hong K (2015) A fainting condition detection system using thermal imaging cameras based object tracking algorithm. J Converg 6(3):1–15
21. Park S, Kim S, Ha Y (2016) Highway traffic accident prediction using VDS big data analysis. J Supercomput 72(7):2832–2832
22. Mazhar Rathore M, Ahmad A, Paul A (2016) Real time intrusion detection system for ultra-high-speed big data environments. J Supercomput 72(9):3489–3510
23. Kim D-H, Choi K-H, Li K-J, Lee Y-S (2015) Performance of vehicle speed estimation using wireless sensor networks: a region-based approach. J Supercomput 71(6):2101–2120