International Journal of Linguistics and Computational Applications (IJLCA)
Volume 2, Issue 1, January - March 2015
ISSN 2394-6385 (Print)
ISSN 2394-6393 (Online)

GPS-Based Smart System for Enhancing Driving Directions for Finding Fastest Route using Driver's Intelligence

R.Loganathan[#1], N.Vivekananda Moorthy *[2]

[1]*Dept. of Information Technology, SRM University, Chennai, India.*
loganathanrajappan@gmail.com
[2] *Asst. Professor, Dept. of Information Technology, SRM University, Chennai, India.*
vivekanthamoorthy.n@ktr.srmuniv.ac.in

Abstract— Traveling is a part of every person's day-to-day life. With the massive and complicated road network of a modern city or country, finding a good route to travel from one place to another is not a simple task. The knowledge of the actual current state of the road traffic and its short-term and dynamic path evolution for the entire road network is a basic component of ATIS (Advanced Traveler Information Systems) and ATMS Advanced Traffic Management System) applications. In this view the use of real-time Taxi Data (TD), based on traces of GPS positions to gather accurate travel times/speeds in a road network and to improve short-term predictions of travel conditions.

GPS-equipped taxis can be regarded as traffic flows on road surfaces, and taxi drivers are usually experienced in finding the fastest (quickest) route to a destination based on their knowledge. We mine smart driving directions from the historical GPS trajectories of a large number of taxis, and provide a user with the practically fastest route to a given destination at a given departure time. In our approach, we propose a time-dependent landmark graph, where a node (landmark) is a road segment frequently traversed by taxis, to model the intelligence of taxi drivers and the properties of dynamic road networks. The essential components that will be discussed are a Web-services-based data collection approach then, a Variance-Entropy-Based Clustering approach is devised to estimate the distribution of travel time between two landmarks in different time slots. Based on this graph, we design a two-stage routing algorithm to compute the practically fastest route. In our existing system static (Dynamic)-path and not update the rout.

Keywords— *Data mining, Spatial databases, Driving directions, time-dependent fast route, taxi trajectories, T-Drive, landmark graph*

1. Introduction

Finding efficient driving directions has become a daily activity and been implemented as a key feature in many map services like Google and Bing Maps. A fast driving route saves not only the time of a driver but also energy consumption (as most gas is wasted in traffic jams). In practice, big cities with serious traffic problems usually have a large number of taxis traversing on road surfaces. These taxis have already been embedded with a GPS sensor, which enables a taxi to report on its present location to a data center in a certain frequency. Thus, a large number of time-stamped GPS trajectories of taxis have been accumulated and are easy to obtain. Intuitively, taxi drivers are experienced drivers who can usually find out the fastest route to send passengers to a destination based on their knowledge (we believe most taxi drivers are honest although a few of them might give passengers a roundabout trip). When selecting driving directions, besides the distance of a route, they also consider other factors, such as the time-variant traffic flows on road surfaces, traffic signals and direction changes contained in a route, as well as the probability of accidents. These factors can be learned by experienced drivers but are too subtle and difficult to incorporate into existing routing engines.

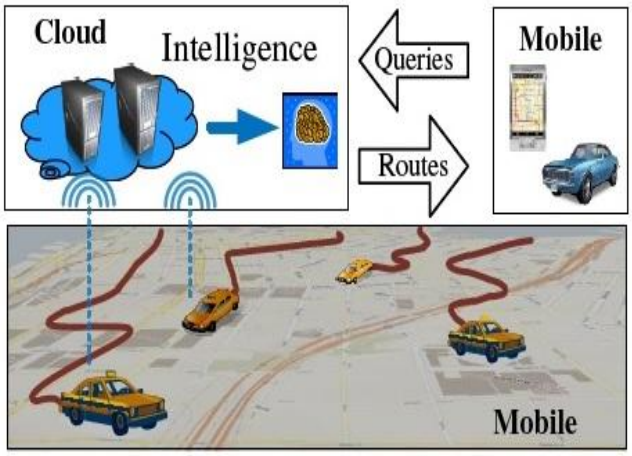


Fig. 1: A cloud-based driving directions service

Therefore, these historical taxi trajectories, which imply the intelligence of experienced drivers, provide us with a valuable resource to learn practically fast driving directions. We propose to mine smart driving directions from a large number of real-world historical GPS trajectories of taxis.

As shown in Figure 1, taxi trajectories are aggregated and mined in the Cloud to answer queries from ordinary drivers or Internet users. Given a start point and destination, our method can suggest the practically fastest route to a user according to his/her departure time and based on the intelligence mined from the historical taxi trajectories. As the taxi trajectories are constantly updated in the Cloud, the suggested routes are state-of-the-art. When proposing the above-mentioned strategy, two major concerns come to people's minds.

First, some routes on which a taxi can quickly traverse might not be feasible for normal drivers, e.g., the carpool tracks in some highways and a few bus-taxi-preserved tracks in a city. But, in most cases, especially in many urban cities like New York and Beijing, private cars can share the same tracks with taxis. That is, taxis' trajectories can still be referenced by other drivers when finding driving directions in an urban city.

Second, the historical trajectory-based approach might not be agile enough to handle some urgent accidents in contrast to real-time traffic analysis. The traffic flows of a city follow some patterns unless some emergent events happen, such as serious accidents, traffic control and road-works. Given that the probability of these events is much lower than that of regular traffic patterns, our method is still very useful in most situations.

At the same time, besides the traffic flow, our method also implicitly incorporates additional factors, such as direction changes and traffic signals. Moreover this method can find the fastest route in a future time and needs less online communication for data transition. Thus, our solution and the real-time-based approach can complement each other. We need to face the following three challenges.

1.1. Intelligence Modelling

As a user can select any place as a source or destination, there would be no taxi trajectory exactly passing the query points. That is, we cannot answer user queries by directly mining trajectory patterns from the data. Therefore, how to model taxi drivers' intelligence that can answer a variety of queries is a challenge.

1.2. Data Sparseness and Coverage

We cannot guarantee there are sufficient taxis traversing on each road segment even if we have a large number of taxis. That is, we cannot accurately estimate the speed pattern of each road segment.

1.3. Low-sampling-rate Problem

To save energy and communication loads, taxis usually report on their locations in a very low frequency, like 2-5 minutes per point. This increases the uncertainty of the routes traversed by a taxi. As shown in Figure 2, there could exist four possible routes ($R1$-$R4$) traversing the sampling points a and b.

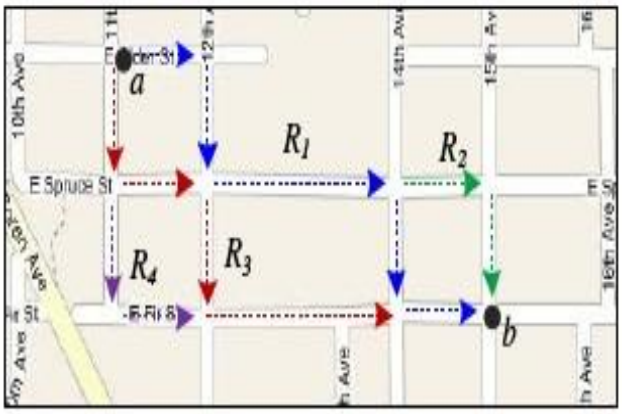


Fig. 2: Low-sampling-rate problem

In our approach, we model a large number of historical taxi trajectories with a time-dependent landmark graph, in which a node (landmark) represents a road segment frequently traversed by taxis. Based on this landmark graph, we perform a two-stage routing algorithm that first searches the landmark graph for a rough route (represented by a sequence of landmarks) and then finds a refined route sequentially connecting these landmarks.

2. System Overview

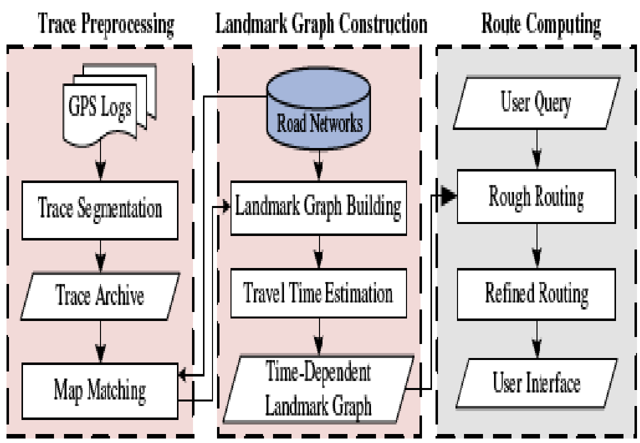


Fig. 3: Architecture overview

The architecture of our system consists of three major components:

- Trajectory Preprocessing,
- Landmark Graph Construction
- Route Computing

The first two components operate offline and the third is running online. The offline parts only need to be performed once unless the trajectory archive is updated.

3. Trajectory Preprocessing

3.1 Trajectory segmentation

In practice, a GPS log may record a taxi's movement of several days, in which the taxi could send multiple passengers to a variety of destinations. Therefore, we partition a GPS log into some taxi trajectories representing individual trips according to the taximeter's transaction records. There is a tag associated with a taxi's reporting when the taximeter is turn on or off, i.e., a passenger gets in or out of the taxi.

3.2 Map matching

To map-match each GPS point of a trip to the corresponding road segment where the point was recorded. As a result, a taxi trajectory is converted to a sequence of road segments.

3.3 Landmark Graph Construction

We separate the weekday trajectories from the weekend ones, and build a landmark graph for weekdays and weekends respectively. When building the graph, we first select the top-k road segments with relatively more projections (i.e., being frequently traversed by taxis) as the landmarks. Then, we connect two landmarks with a landmark edge if there are a certain number of trajectories passing these two landmarks. Later, we estimate the distribution of travel time of each landmark edge by using the VE-clustering algorithm. Now, a time-dependent landmark graph is ready for online computation.

3.4 Route Computing

Given a query (qs, qd, td), we carry out a two-stage routing algorithm to find out the fastest route. In the first stage, we perform a rough routing that searches the time-dependent landmark graph for the fastest rough route represented by a sequence of landmarks. In the second stage, we conduct a refined routing algorithm, which computes a detailed route in the real road network to sequentially connect the landmarks in the rough route.

4. Problem Definition

4.1 Road Segment

A road segment r is a directed (one-way or bidirectional) edge that is associated with a direction symbol ($r.dir$), two terminal points ($r.s$, $r.e$), and a list of intermediate points describing the segment using a polyline. If $r.dir$=one-way, r can only be traveled from $r.s$ to $r.e$, otherwise, people can start from both terminal points, i.e., $r.s \rightarrow r.e$ or $r.e \rightarrow r.s$. Each road segment has a length $r.length$ and a speed constraint $r.speed$, which is the maximum speed allowed on this road segment.

4.2 Dynamic Road Network

A dynamic road network G_r is a directed graph, G_r = (V_r,), where V_r is a set of nodes representing the terminal points of road segments, and E_r is a set of edges denoting road segments. The time needed for traversing an edge is dynamic at least in the following two aspects:

4.2.1 Time-dependent

Typically, the traffic flow on a road surface varies over days of the week and time of day, e.g., a road could become crowded in rush hours while be quite smooth at other times.

4.2.2 Location-variant

Different roads have different time-variant traffic patterns. For instance, some streets could still be very fast even in the morning rush. However, the rush hours of a few roads may last for a whole day.

4.3 Route

A route R is a set of connected road segments, i.e., : $r_1 \rightarrow r_2 \rightarrow \cdot\cdot\cdot \rightarrow r_n$, where r_{k+1}.s = $r_k.e$, ($1 \leq k < n$). The start point and end point of a route can be represented as $R.s = r_1.s$ and $R.e = r_n.e$.

4.4 Taxi Trajectory

A taxi trajectory T_r is a sequence of GPS points pertaining to one trip. Each point p consists of a longitude, latitude and a time stamp $p.t$, i.e., T : $p_1 \rightarrow p_2 \rightarrow \cdot\cdot\cdot \rightarrow p_n$, where $0 < p_{i+1}.t - p_i.t < \Delta T$ ($1 \leq i < n$). ΔT defines the maximum sampling interval between two consecutive GPS points.

4.5 Landmark

A landmark is one of the top-k road segments that are frequently traversed by taxi drivers according to the trajectory archive.

4.6 Transition

Given a trajectory archive A, a time threshold t_{max}, two landmarks u, v, arriving time t_a, leaving time t_l, we say S = (u, v; t_a, t_l) is a transition if the following conditions are satisfied: (1) There exists a trajectory T_r = (p_1, p_2, . . . , p_n) ∈ A, after map matching, T_r is mapped to a road segment sequence ($r_1, r_2, \ldots, r_n$). ∃ i, j, 1 ≤ i < j ≤ n s.t. $u = r_i$, $v = r_j$.(2) $r_{i+1}, r_{i+2}, \ldots, r_{j-1}$ are not landmarks.(3) $t_a = p_{i.t}$, $t_l = p_{j.t}$ and the travel time of this transition is $t_l - t_a \leq t_{max}$.

4.7 Candidate Edge and Frequency

Given two landmarks u, v and the trajectory archive A, let S_{uv} be the set of the transitions connecting (u, v). If $S_{uv} \neq \emptyset$, we say $e = (u, v; T_{uv})$ is a candidate edge, where $T_{uv} = \{(t_a, t_l)|(u, v; t_a, t_l) \in S_{uv}\}$ records all the historical arriving and leaving times. The support of e, denoted as $e.supp$, is the number of transitions connecting (u, v), i.e., $|S_{uv}|$. The frequency of e is $.supp/T$, denoted as $e.freq$, where τ represents the total duration of trajectories in archive A.

4.8 Landmark Edge

Given a candidate edge e and a minimum frequency threshold δ, we say e is a landmark edge if $e.freq \geq \delta$.

4.9 Landmark Graph

A landmark graph $G_l = (V_l,)$ is a directed graph that consists of a set of landmarks V_l (conditioned by k) and a set of landmark edges E conditioned by δ andt_{max}.

4.10 Optimism Index

The optimism index α indicates how fast a person would like to drive as compared to taxi drivers. The higher rank (position in taxi drivers), the faster the person would like to drive.

4.11 Problem Definition

Given a user query with a start point q_s, a destination q_d and a departure time t_d, find the fastest route R in a dynamic road network $G_r = (V_r, E_r)$ which is learned from a trajectory archive A.

5. Time-Dependent Landmark Graph

The TDLG is we use "landmark" to model the taxi-drivers' intelligence is that: 1) The notion of landmarks follows the natural thinking pattern of people, and can give users a more understandable and memorable presentation of driving directions beyond detailed descriptions. For instance, the typical pattern that people introduce a route to a driver is like this "take I-405 South at NE 4th Street, then change to I-90 at exit 11, and finally exit at Q west Field". Instead of giving turn-by-turn directions, which a driver cannot remember, people prefer to use a few landmarks (like NE 4th Street) that highlight key directions to the destination.2.The sparseness and low-sampling-rate of the taxi trajectories do not support the speed estimation for each road segment while we can estimate the traveling time between two landmarks. Meanwhile, the low-sampling-rate trajectories cannot offer sufficient information for inferring the exact route traversed by a taxi (refer to Figure 2). Thus, we can only use a road segment instead of their terminal points as a landmark. Here, we detect the top-k road segments as the landmarks instead of setting up a fixed threshold, since a threshold will vary in the scale of taxi trajectories.

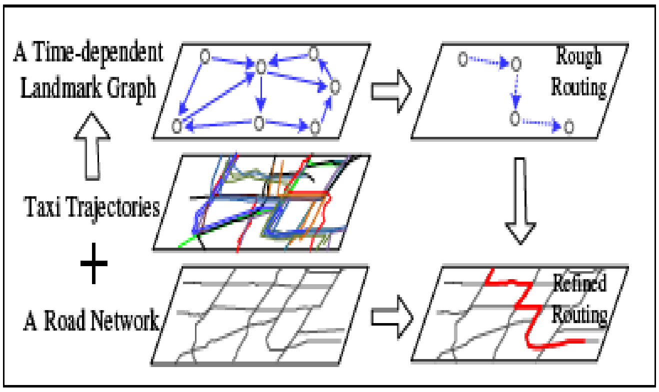


Fig. 4: Hierarchical architecture

5.1 Variance-Entropy-Based Clustering

VE-Clustering algorithm, which is a two-phase clustering method, to learn different time partitions for different landmark edges based on the taxi-trajectories. Given a landmark edge $e = (u, v; T_{uv})$, our goal is to estimate the travel time from u to v based on T_{uv} (T_{uv} is the collection of (t_a, t_l) pairs of e defined in Definition). The travel time of the transitions pertaining to a real landmark edge in a two dimensional space, where the x and y axes denote the arriving time (t_a) and travel time $(t_l - t_a)$ respectively (where $x_i = t_a$, $y_i = t_l - t_a$). As the number of clusters and the boundary of these clusters vary in different landmark edges.

5.2 V-Clustering

We cluster the travel times of transitions pertaining to a landmark edge into several categories based on the variance of these transitions' travel times. We first sort T_{uv} according to the values of travel time $(t_l - t_a)$, and then partition the sorted list L into several sub-lists in a binary-recursive way. In each iteration, we first compute the variance of all the travel times in L. Later, we find the "best" split point having the minimal weighted average variance (WAV) defined as Equation 1:

$$\mathrm{WAV}(i; L) = \frac{|L_1(i)|}{|L|}\mathrm{Var}(L_1(i)) + \frac{|L_2(i)|}{|L|}\mathrm{Var}(L_2(i))$$

Where $L1(i)$ and $L2(i)$ are two sub-lists of L split at the i_{th} element and Var represents the variance. This best split point leads to a maximum decrease of Equation 2:

$\Delta V(i) = \mathrm{Var}(L) - \mathrm{WAV}(i;L)$

The algorithm terminates when $\max\{\Delta V(i)\}$ is less than a threshold. As a result, we can find out a set of split points dividing the whole list L into several clusters $C = \{c_1, c_2, \ldots, c_m\}$, each of which represents a category of travel times. the travel times of the landmark edges have been clustered into three categories plotted in different colors and symbols.

5.3 E-Clustering

This step aims to split the x-axis into several time slots such that the travel times have a relatively stable distribution in each slot. After V-Clustering, we can represent each travel time y_i with the category it pertains to $((y_i))$, and then sort the pair collection $S^{xc} = \{(x_i, c(y_i))\}^{n}_{i=1}$ according to x_i (arriving time). The information entropy of the collection S^{xc} is given by:

$$\mathrm{Ent}(S^{xc}) = -\sum_{i=1}^{m} p_i \log(p_i)$$

where p_i is the proportion of a category c_i in the collection. The E-Clustering algorithm runs in a similar way to the V-Clustering to iteratively find out a set of split points. The only difference between them is that, instead of the WAV, we use the weighted average entropy of S^{xc} defined as:

$$\mathrm{WAE}(i; S^{xc}) = \frac{|S_1^{xc}(i)|}{|S^{xc}|}\mathrm{Ent}(S_1^{xc}(i)) + \frac{|S_2^{xc}(i)|}{|S^{xc}|}\mathrm{Ent}(S_2^{xc}(i))$$

In the E-Clustering, where $S_1{}^{xc}$ and $S_2{}^{xc}$ are two subsets of S^{xc} when split at the i_{th} pair. The best split point induces a maximum information gain which is given by

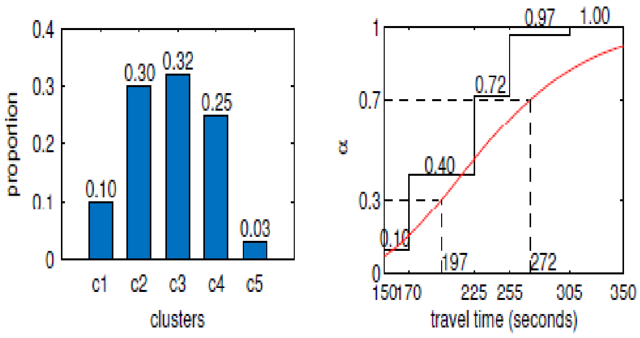
$$\Delta(i) = \mathrm{Ent}(S^{xc}) - \mathrm{WAE}(i; S^{xc})$$

We can compute the distribution of the travel times in each time slot after the E-Clustering process.

5.4 Route computing

The traffic condition of a road, the travel time of a route also depends on drivers. Sometimes, different drivers take different amounts of time to traverse the same route at the same time slot. For example, people familiar with a route can usually pass the route faster than a new-comer. Also, even on the same path, cautious people will likely drive relatively slower than those preferring to drive very fast and aggressively. To catch the above factor caused by individual drivers, we define the optimism index α.For example, α = 0.9 means a person usually drives as fast as the top 10% (i.e., 1-0.9) fast- driving taxi drivers. α = 0.2 means that drivers can only outperform the bottom 20% of taxi drivers. The α can be learned from a driver's historical trajectories or set by them.

Given a user's optimism index α, we can determine his/her time cost for traversing a landmark edge e in each time slot based on the learnt travel time distribution. For example, Figure 5(a) depicts the travel time distribution of an landmark edge in a given time slot ($c1 \sim c5$ denotes 5 categories of travel times). Then, we convert this distribution into a cumulative frequency distribution function and fit a continuous cumulative frequency curve shown in Figure 5.



a)Travel time distribution (b) Cumulative frequency

Fig. 5: Optimism index

This curve represents the distribution of travel time in a given time slot. That is, the travel times of different drivers in the same time slot are different. So, we cannot use a single-valued function. For example, given α=0.7, we can find out the corresponding travel time is 272 seconds, while if we set α=0.3 the travel time becomes 197 seconds.

6. Evaluation

6.1 Data - Setting

6.1.1 Road Network

We perform the evaluation based on the road network of Tamil Nadu, which has 14,257 road nodes and 37,380 road segments.

6.2 Taxi Trajectories

We build our system based on a real trajectory dataset generated by taxis. The total distance of the data set is more than 1,99,040 thousand kilometers The average sampling interval of the data set is 3.1 minutes per point and the average distance between two consecutive points is about 600 meters.

6.3 Evaluating landmark graphs

We build a set of landmark graphs with different values of k ranging from 500 to 13000. The threshold δ is set to 10, i.e., at least ten times per day traversed by taxis and t_{max} is set to 30 minutes. Our project each real-user trajectory to our time-dependent landmark graph, and use the landmark graph to estimate the travel time of the trajectory. We study the accuracy of the time estimation changing over k and α. We also investigate the accuracy changing over the scale of the taxi trajectory dataset.

6.4 Evaluation based on synthetic queries

We generate 500 queries with different TN-distances and departure times. The TN-distances between the start point and destination ranges from 3 to 43km and follows a uniform distribution. The departure time ranges from 6am to 10pm and was generated randomly in different time slots.

6.5 In-the-field evaluation

We conduct two types of in the field studies:1) The same driver traverses the routes suggested by our method and baselines at different times. 2) Two drivers (with similar driving skills and habits) travel different routes (recommended by different methods) simultaneously.

7. Related Work

7.1. Driving Direction Services

Our work differs from the existing routing services as follows. First, our driving direction service considers the factor a user, and automatically adapts to the user's driving behavior according to his/her driving paths. Second,we model the historical traffic pattern using the landmark graph, and integrate this information into a time-dependent routing algorithm. Third, we mine drivers' intelligence from taxi trajectories. The intelligence is far beyond the route distance and traffic flows.

7.2. Time-Dependent Fastest Path

The time-dependent fastest path problem is first considered in paper[8]. Dreyfus [9] suggested a straightforward generalization of Dijkstra algorithm but the authors did not notice it does not work for a non-FIFO network. Under the FIFO assumption, paper [14] provides a generalization of Dijkstra-algorithm that can solve the problem with the same time complexity as the static fastest route problem. Demiryurek. Present a good case study comparing existing approaches for the TDFP problem on real-world networks.

7.3. Traffic-Analysis-Based Approach

As a very complex problem, urban traffic flow analysis has been studied based on the readings of road sensors and floating-car-data. These works follow the paradigm of "sensor data → traffic flow → drives direction", and are useful in detecting unexpected traffic jams and accidents. The major challenge of such kinds of solutions is the small coverage and sparse density of the sensor data. For example, the traversing speed of a highway with enough road sensors or floating cars can be accurately estimated, while the inferred speed of many service roads, streets and lanes (without enough sensors) are not that precise[14]. Given that users can select any locations as destinations, sometimes the path finding algorithms based on the inferred real-time traffic might not perform as well as we expects. Different from the above methods, our approach is based on many taxi drivers' intelligence mined from their historical trajectories. This intelligence has implied all the key factors finding a fast driving route. Actually, GPS-embedded taxis can be regarded as probing real-time traffic on roads, and the accumulated historical GPS trajectories reflect the long-term traffic patterns of a city. As the traffic flows of a city follow some patterns in most cases, our method is very valuable in finding practically fast driving routes for users.

8. Conclusion

This project designed a framework for finding fastest route from source and destination. Time Dependent landmark graph algorithm is designed to provide the user with dynamic route when the traffic is high and designed for eliminating the round trip travel by analysing the landmarks in the route and then Variance Entropy Algorithm is designed for determining the estimated travel time between two places. Finds out the practically fastest route to a destination at a given departure time in terms of taxi drivers' intelligence learned from a large number of historical taxi trajectories.

In our method, we first construct a time-dependent landmark graph, and then perform a two-stage routing algorithm based on this graph to find the fastest route. We build a real system with real world GPS trajectories generated by taxis evaluate the system with extensive experiments and in-the-field evaluations. The results show that our method significantly outperforms both the speed constraint-based and the real-time-traffic-based method in the aspects of effectiveness and efficiency. Given over 5 taxis in a region of 1km, more than 60% of our routes are faster than that of the speed-constraint-based approach, and 50% of these routes are at least 20% faster than the latter. On average, our method can save about 16% of time for a trip, i.e., 5 minutes per 30-minutes driving. We agree that a recommended route would become crowded if many people take it. This is the common problem of path-finding, and this problem is even worse (than ours) in present shortest-path and real-time-traffic-based methods (as our method can be customized for different drivers). In the future, we can reduce this problem by using some strategies, such as load balance (offer top three routes) and data update (in a relatively fast frequency). Another direction in which we are going to move forward is combining real-time traffic information with our approach.

References

[1] J. Yuan, Y. Zheng, C. Zhang, W. Xie, G. Sun, H. Yan, and X. Xie,“ T-Drive: Enhancing Driving Directions with Taxi Drivers' Intelligence,” Proc.18th SIGSPATIAL Int'l Conf. Advances in Geographic Information Systems (GIS), 2010.

[2] T. Hunter, R. Herring, P. Abbeel, and A. Bayen, “Path and Travel Time Inference from GPS Probe Vehicle Data,” Proc.Neural Information Processing Systems (NIPS), 2009.

[3] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang,“Map-Matching for Low-Sampling-Rate GPS Trajectories,” Proc.Int'l Conf. Advances in Geographic Information Systems (GIS),2009.

[4] J. Yuan, Y. Zheng, C. Zhang, and X. Xie, “An Interactive-Voting Based Map Matching Algorithm,” Proc. Int'l Conf. Mobile Data Management (MDM), 2010.

[5] E. Kanoulas, Y. Du, T. Xia, and D. Zhang, “Finding Fastest Paths on a Road Network with Speed Patterns,” Proc. Int'l Conf. Data Eng. (ICDE), 2006.

[6] Y. Zheng, L. Liu, L. Wang, and X. Xie, “Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web,” Proc. 17th Int'l Conf. World Wide Web (WWW),2008.

[7] C. de Fabritiis, R. Ragona, and G.Valenti, “Traffic Estimation and Prediction Based on Real Time Floating Car Data,” Proc. 11th Int'l IEEE Conf.Intelligent Transportation Systems (ITSC '08), pp. 197-203, Oct. 2008.

[8] U. Demiryurek, F. Banaei-Kashani, and C. Shahabi, “A Case for Time-Dependent Shortest Path Computation in Spatial Networks,” Proc. Int'l Conf. Advances in Geographic Information Systems (GIS), 2010.

[9] A.Thiagarajan, L.Ravindranath, K.LaCurts, S.Madden, H.Balakrishnan, S.Toledo, and J. Eriksson, “Vtrack: Accurate, Energy-Aware Road Traffic Delay Estimation Using Mobile Phones,” Proc. Seventh ACM Conf. Embedded Networked Sensor Systems, 2009.

[10] A. Bejan, R. Gibbens, D. Evans, A.Beresford, J. Bacon, and A.Friday, “Statistical Modelling and Analysis of Sparse Bus Probe Data in Urban Areas,” Proc. Int'l IEEE Conf. Intelligent Transportation Systems (ITS), 2010.

[11] D. Pfoser, S. Brakatsoulas, P. Brosch, M. Umlauft, N. Tryfona, and G. Tsironis, “Dynamic Travel Time Provision for Road Networks,”Proc. ACM SIGSPATIAL Int'l Conf. Advances in Geographic Information Systems (GIS), 2008.

[12] B. Ziebart, A. Maas, A. Dey, and J. Bagnell, “Navigate Like a Cabbie: Probabilistic Reasoning from Observed Context-Aware Behavior,” Proc. Int'l Conf. Ubiquitous Computing, 2008.

[13] H. Gonzalez, J. Han, X. Li, M.Myslinska, and J. Sondag,“Adaptive Fastest Path Computation on a Road Network: A Traffic Mining Approach,” Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB), 2007.

[14] H.A. Karimi and X. Liu, “A Predictive Location Model for Location-Based Services,” Proc. ACM Int'l Conf. Advances in Geographic Information Systems (GIS), pp. 126-133, 2003.

[15] B.C. Dean, “Continuous-Time Dynamic Shortest Path Algorithms,”master's thesis, MIT, 1999.

[16] S.-W. Kim, J.-I. Won, J.-D. Kim, M. Shin, J. Lee, and H. Kim, “Path Prediction of Moving Objects on Road Networks through Analyzing Past Trajectories,” Proc. 11th Int'l Conf. Knowledge-Based and Intelligent Information and Eng. Systems (KES), pp. 379-389,2007.

[17] N. Malviya, S. Madden, and A.Bhattacharya, “A Continuous Query System for Dynamic Route Planning,” Proc. Int'l Conf. Data Eng. (ICDE), pp. 792-803, 2011.

[18] J. Letchner, J. Krumm, and E. Horvitz, “Trip Router with Individualized Preferences (trip): Incorporating Personalization into Route Planning,” Proc. Conf. Innovative Applications of Artificial Intelligence (NCAI), 2006.

[19] B. Liu, “Route Finding by Using Knowledge about the Road Network,” IEEE Trans. Systems, Man, and Cybernetics, Part A:Systems and Humans, vol. 27, no. 4, pp. 436-448, July 1997.

[20] Z. Chen, H.T. Shen, and X. Zhou, “Discovering Popular Routes from Trajectories,” Proc. Int'l Conf. Data Eng. (ICDE), pp. 900-911, 2011.

R.Loganathan, MTech in Department of Information Technology at SRM University, Chennai. He holds a Bachelor Degree in Computer Science from AVS Engineering College (Under Anna University, Chennai).

N. Vivekananda Moorthy is an Assistant Professor of Department of Information Technology, at SRM University, Chennai. He holds a Master Degree in Computer Science and Engineering from the Bharathidasan University. In addition he is a Researcher in E-Learning. He has published many articles in the National and International Journals of Computer Science and presented papers in many Conferences and having 8 years of teaching experience.