Enhancing Driving Direction Time based on Speed Fluctuation and Vehicle Type Identification

A. T.Vijayakumar, B. T sripriya
Asst.Professor(Sr.Grade) Department of Information Technology Bannari Amman Institute of Technology, Sathyamangalam,India
vijayakumart@bitsathy.ac.in
PG Scholar Department of Information Technology Bannari Amman Institute of Technology, Sathyamangalam,India
sripriyat.se12@bitsathy.ac.in

Abstract
Finding efficient driving directions has become a daily activity and been implemented as a key feature in many map services like WMS, Google and Bing Maps. A fast driving route saves the time of a driver and also energy consumption (as most gas is wasted in traffic jams). GPS-equipped taxis can be regarded as mobile sensors probing 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 the 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.

1) Vehicle detection, and tracking (2) Feature extraction and classification (3) Database storage and retrieval
Driver behavior based on surrounding vehicles, as well as surveillance from mobile platform. 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. We build our system based on a real world trajectory simulation dataset and stored in MOD.

Keywords: Spatial databases and GIS, data mining, GPS trajectory, driving directions, driving behavior, speed fluctuation, vehicle type

I. INTRODUCTION
Find an efficient driving directions has become a daily activity and have 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). Therefore, this service is important for economy of both end users and governments aiming to ease traffic problems and protect environment.

Essentially, the time that a driver traverses a route depends on the following five aspects: 1) the physical feature of a route, such as distance, capacity (lanes), and the number of traffic lights as well as direction turns; 2) the time-dependent traffic flow on the route; and 3) a user’s driving behavior. Given the same route, cautious drivers will likely drive relatively slower than those preferring driving very fast and aggressively. Also, users’ driving behaviors usually vary in their progressing driving experiences. For example, traveling on an unfamiliar route, a user has to pay attention to the road signs, hence drive relatively slowly. Thus, a good routing service should consider these three aspects (routes, traffic, and drivers), which are far beyond the scope of the shortest/fastest path computing. 4) the speed of vehicle, speed variations is used to identify trajectory of the road. 5) the type of vehicle gives information to choose lane.
Usually, big cities have a large number of taxicabs traversing in urban areas. For efficient taxi dispatching and monitoring, taxis are usually equipped with a GPS sensor, which enables them to report their locations to a server at regular intervals, e.g., 2-3 minutes. 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. These factors can be learned by experienced drivers but are too subtle and difficult to incorporate into existing routing engines. Therefore, these historical taxi trajectories, which imply the intelligence of experienced drivers, provide us with a valuable resource to learn practically fast driving directions.

This proposed system a cloud-based to find out the practically fastest route for a particular user at a given departure time. Specifically, the system mines the intelligence of experienced drivers from a large number of taxi trajectories and provide the end user with a smart route, which incorporates the physical feature of a route, the time-dependent traffic flow as well as the users’ driving behaviors (of both the fleet drivers and of the end user for whom the route is being computed). We build a real system with real-world GPS trajectories generated by over 33,000 taxis in a period of three months, then evaluate the system with extensive experiments and in-the-field evaluations. Our method significantly better performance than the competing methods in the aspects of effectiveness and efficiency in finding the practically fastest routes. Overall, more than 60 percent of our routes are faster than that of the existing online map services, and 50 percent of these routes are at least 20 percent faster than the latter.

A cloud-based driving directions service. As shown in Figure 1, the taxi trajectories are collected and mined in the cloud to answer queries from the internet users.

**Figure 1.** A cloud-based driving directions service

*Preprocessing steps:*

To face the following challenges:

1) Intelligence modeling: User can select any place as a source or destination, we can’t answer users query by directly mining trajectory pattern from the data. So model taxi drivers intelligence that can answer variety of queries (yuan et al, 2013) 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 (Yuan et al., 2013) (Yu Zheng et al., 2010) 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 (Yuan et al., 2010).

II. RELATED WORK

In the previous paper (Yuan et al., 2010), we propose the notion of a time-dependent landmark graph, which well models the intelligence of taxi drivers based on the taxi trajectories. We devise a Variance-Entropy-Based Clustering (VE-Clustering for short) method to learn the time-variant distributions of the travel times between any two landmarks.

In the previous paper (Yuan et al., 2013), we further improve our routing service by self-adaptively learning the driving behaviors of both the taxi drivers and the end users so as to provide personalized routes to the users (Yuan et al., 2013). We present smoothing algorithms for removing the round about part of the original rough routes. We build the improved system by using a real-world trajectory data set generated by 33,000 taxis in a period of three months, and evaluate the system by conducting both synthetic experiments and in-the-field evaluations (performed by real drivers). The results show that proposed method can effectively and efficiently find out practically better routes than the competing methods. We further improve the fastest route by calculating the speed fluctuation, changes in the shaft speed during a cycle.

Average/nominal shaft speed \( \omega_{\text{avg}} \)

\[
\text{FI} = \omega_{\text{max}} - \omega_{\text{min}}
\]

Co-efficient of speed fluctuation

\[
C_f = \frac{\omega_{\text{max}} - \omega_{\text{min}}}{\omega_{\text{avg}}}
\]

In the proposed traffic monitoring approach, objects and its position will be change periodically due to dynamic and mobility nature. There should be a provision to monitor behavior and position of the on the regular basis. This will help analyze the traffic flow and object behavior periodically. Based on the object movement on every region the position will be updated. Based on the position and time the speed can be calculated. The frequency of speed will helps to identify the traffic situation of the region. Instead of monitoring individual node movement and speed the system will analyze region based multi object speed analysis.

That also defines minimum and maximum speed and average speed flow of the object in a particular region. After pausing for a certain period of time, the object can selects a new random path based on the distance and maximum speed path.

In this process a novel Position-based speed updating process is proposed, in which several moving objects recognized by the system and the data will be analyzed for further classification. Vehicle type can be identified through the GPS id, while registering GPS vehicle information also uploaded. Speed of the vehicle can track from this type.

III. PROBLEM DEFINITION

Road Segments:

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

**Road Network:**

A road network \( r_n \) is a directed graph, which consists of vertices and edges of the road segments \( r_{on} \), \( V_r, E_r \). \( V_r \) is a set of nodes representing the terminal points of road segments and \( E_r \) is a set of edges denoting the road segments. The time varied based on the day traffic.

**Route:**

A route \( R \) is a set of connected road segments, \( r_1, r_2, \ldots, r_n \), where the starting point of the road as source \( R_s = r_{1,s} \) and destination \( R_e = r_{n,e} \).

**Speed Fluctuation:**

A variation in the engine speed or vehicle speed fluctuation due to the increase of torque is restrained. Then the fluctuation of the vehicle speed in the retreat into a garage or the like is restrained to facilitate the drive of the vehicle.

**IV. OVERVIEW**

As shown in figure 2, the architecture of our system consists of three major components: Moving data Analysis, Trajectory Calculation, Route Computing.

![Overall architecture diagram](image)

**Figure 2.** Overall architecture diagram

**A. Moving data Analysis**

This will divide into two segments, first GPS trajectories are converted into trips then matches each trip with road networks. 1) Trajectory segmentation: GPS log may record a variety of taxi’s moving on several days, in which the taxis can send multiple persons to multiple destinations. We partition the log into individual trips according to the taximeter’s records. A tag is associated with vehicle reporting when the vehicle is turned on or off. 2) Map matching: Map-matching is the process of aligning a sequence of observed user positions with the road network on a digital map. It is a preprocessing step for many applications, such as moving object management, traffic flow analysis, and driving directions. In practice there exists huge amount of low-sampling-rate (e.g., one point every 2-5 minutes) GPS trajectories.

**B. Trajectory Calculation:**
Data structure type and its algorithms are needed for implementing the operations. Selection and join are needed for spatial operation. Specialized index structures are needed for selection operation. R-tree is used to organize in hierarchical set of rectangles. Temporal state relation and SQL queries are used to store and retrieve data from database.

Trajectory indexing by spatial temporal R-tree: R-tree is also balanced search tree, organize the data in pages and storage in disk.R-tree is suitable for large dataset and databases. Data is organized in R-tree; many algorithms are used for accessing the data from R-tree.A spatial data are accessed on the k nearest neighbors’ algorithms.

C. Route Calculation:

Two-stage routing algorithm to find out the fastest route: Given a query (qs, qd, td), 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.

**Rough routing landmark graph:**

Based on the traffic condition of the road, the travel time of vehicle depends on drivers. Travel time will be varying even the same route at the same time slot. Based on driving habit, skills and familiarity of routes experienced driver with familiar route can travel faster than inexperienced driver.

In the rough routing, we first search n (in our system, we set n=2) nearest landmarks for qs and qd, respectively (a spatial index is used), and formulate n X n pair of landmarks. For each pair of landmarks, we find the time dependent fastest route on the landmark graph by using the Label-Setting algorithm (Dean, 2009) (Lou et al., 2009), which is a generalization of Dijkstra algorithm. For any visited landmark edge, we use the custom factor to determine the travel time. The time costs for traveling from qs and qe to their nearest landmarks are estimated in terms of speed constraint. For example, in Fig. 4, if we start at time td = 0, the fastest route from qs to qd is qs → r3 → r4 → qd. When we arrive at r3, the time stamp is 0.5, the travel time of e34 is 1, and then the total time of this route is 0.5+1+0.5= 2. However, if we start at td =2, the route qs → r1 → r2 → qd now becomes the fastest rough route since when we arrive at r3, the travel time of the e34 becomes and the total time of the previous route is now 3.

**Refined Routing:**

Even using the state-of-the-art map-matching algorithm, the accuracy is less than 70 percent (Yuan et al., 2010) for the low-sampling rate trajectories. For example, as shown in Fig. 3, r2 and r4 are wrongly mapped road segments, the actual route is along the horizontal road from qs to qd. The map-matching error results in that r2 and r4 are recognized as landmarks and brings noise when estimating the travel time, e.g., the real travel time for r2→r3 is very likely to be much longer than the estimated time due to the map-matching error, which leads to r2 → r3 becomes a part of this rough route.
V. GPS-BASED VEHICLE DETECTION APPROACH

This section outlines the method used in the intelligence layer application for detecting the location of traffic incidents causing congestion. It is assume that each vehicle on the monitored road is equipped with GPS receiver and data transmission device for example GPRS enable mobile device. The GPS data from all the vehicles is transmitted to the server in real-time. The incident detection process involves two phases:

Phase 1: Analyzing Traffic Pattern

To analyze the traffic patterns more efficiently, the roads are dynamically divided into smaller segments. The length of these segments depends on the type of road, date, time and weather conditions. After the segmentation each road segment is assigned a normal average speed range for example, 500 meters segments for a motorway type on weekday’s peak time with normal average speed between 50-70 mph.
under normal weather conditions. The boundary of a segment is represented as geo-fence which is the set of coordinates creating a polygonal layer around the road segment. Each segment has upstream and downstream representing the flow of the vehicles. The vehicles’ current location (coordinates) appearing within the segment boundary (Geo-Fence) determines the segment within which they are currently positioned.

**TABLE I**

**PHASE I ALGORITHM**

<table>
<thead>
<tr>
<th>Analyse traffic patterns on road R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1.1: Segment the road dynamically and assign normal average speed.</td>
</tr>
<tr>
<td>Step 1.2: Calculate in each segment the average speed of vehicles going in a certain direction.</td>
</tr>
<tr>
<td>Step 1.3: Compare current average speed with the normal average speed for the segment. If the average speed for the segment is significantly slower than normal average speed then that segment is marked for further analysis and step 1.4 applies.</td>
</tr>
<tr>
<td>Step 1.4: Identify road segment with the lowest average speed and apply next step.</td>
</tr>
</tbody>
</table>

Phase 1 consists of the following steps:

*Step 1.1:* Segment the road dynamically and assign normal average speed.

*Step 1.2:* Calculate in each segment the average speed of vehicles going in a certain direction.

*Step 1.3:* Compare current average speed with the normal average speed for the segment. If the average speed for the segment is significantly slower than normal average speed then that segment is marked for further analysis and step 1.4 applies.

*Step 1.4:* Identify road segment with the lowest average speed and apply next step.
Step 1.5: Compare the average speed of the vehicles in the neighboring road segments with that of the marked Segment.

Step 1.6: Determine the current average speed of the road segments in front and behind of the marked segment, see

Step 1.7: Divide the marked segment in Step 1.4 in 10 smaller sub-segments and repeat the steps 1.2 to 1.6 this time for each sub-segment.

In Step 1.5, the average speed in segments in front and behind the marked segment is found in order to detect if there is an incident in the slowest marked segment or there is just a general congestion. If there is just a general congestion, the average speed will be similar in adjoining segments. However, if there is an incident causing a blockage in the market segment then the average speed of the segments in front of that blocked/marked segment will be higher and with less number of vehicles.

Phase 2: Identifying Vehicle Behavior

The GPS data of the vehicles within the identified sub-segment in phase 1 are analysed in the following steps to identify behavior of individual vehicle and to examine if an incident may have occurred.

Step 2.1: Identify the vehicle(s) which have:
- Significantly lower speed than the current average on particular sub-segment
- Completely stopped
- Heading in a direction different to the traffic flow

Step 2.2: Determine from the map data if the segment consists of or is close to a place, where vehicles usually stops e.g. traffic lights, junction or level crossing. If it is, then go to step 2.3, if not go to step 2.4.

Step 2.3: To enable short or temporary blockages to resolve themselves, pause for a time based on the location e.g. for traffic lights wait for 1 minute, for level crossing wait for 5 minutes. If the condition(s) identified in step 2.1 still exist go to step 2.4.

Step 2.4: Analyze the GPS data to determine if:
- All current vehicles’ average speed within the segment became significantly lower than the average speed.
- Vehicle(s) stopped abnormally by analyzing their velocities over time (e.g. a vehicle decelerating from 70mph to 5mph in 2-3 seconds on a motorway).
- The coordinates of a corner of a vehicle are less than 2m from the bounding box of another vehicle and if the two vehicles are separated from each other by less than 5m in height. (It is to avoid level issues such as an overhead bridge with same flow of vehicle direction).

Step 2.5: Pause for a certain time (based on type of road and condition) to check if any change in average speed occurs, if not the system triggers an alarm (notifies the notification service) with the concerning vehicles and the exact incident location of the sub-segment.
VI. EVALUATION

Road network:
We perform the evaluation based on the road network, which consists of 106,579 road nodes and 141,380 road segments.

Taxi trajectories:
We build our system based on a real trajectory data set generated by over 33,000 taxis over a period of three months. The total distance of the data set is more than 400 million kilometers and the total number of GPS points reaches 790 million. 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. After completing the preprocessing, we get a trajectory archive containing 4.96 million trajectories.

Real-user trajectories:
We use the driving history (ranging from two months to one year) of 30 real drivers recorded by GPS loggers to evaluate travel time estimation. This data are a part of the released GeoLife data set (Zheng et al, 2008) (Zheng et al, 2010), and the average sampling interval is about 10s. That is, we can easily determine the exact road segments a driver traversed and corresponding travel times.

A. Evaluation landmark graphs
We build a set of landmark graphs with different values of k ranging from 500 to 15000. The threshold is set to 10, i.e., at least ten times per day traversed by taxis (in total over 900

---

**TABLE II**

**PHASE II ALGORITHM**

<table>
<thead>
<tr>
<th>Identify vehicle behaviour in sub-segment S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: For 1:N // N = number of cars</td>
</tr>
<tr>
<td>IF Speed (N) &lt;&lt; Normal_average_speed</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Mark Vehicle_N</td>
</tr>
<tr>
<td>2: IF ( Normal_stoppage_pts (S) == 0)</td>
</tr>
<tr>
<td>Goto 3 Else Goto 4</td>
</tr>
<tr>
<td>3: Wait for interval T1</td>
</tr>
<tr>
<td>IF no change Goto 4</td>
</tr>
<tr>
<td>4: For Each Vehicle_N in S</td>
</tr>
<tr>
<td>Analyse GPS Data (Vehicle_N)</td>
</tr>
<tr>
<td>// abnormal deceleration, direction change, collision etc. (See 2.4)</td>
</tr>
<tr>
<td>5: Wait for interval T2</td>
</tr>
<tr>
<td>IF no change</td>
</tr>
<tr>
<td>Alert and return (Location coordinates, Vehicles_List)</td>
</tr>
</tbody>
</table>
times in a period of 3 months) $\Delta$ and $t_{\text{max}}$ is set to 30 minutes. We 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 $\tau$. We also investigate the accuracy changing over the scale of the taxi trajectory dataset.

### B. Evaluation Based on Queries

For evaluating the effectiveness of the routes suggested by different methods (say methods A and B), we use the following two criteria: Fast Rate 1 (FR1) and Fast Rate 2 (FR2) where method B is used as a baseline.

$$\text{FR1} = \frac{\text{Number (A’s travel time < B’s travel time)}}{\text{Number(queries)}}$$

$$\text{FR2} = \frac{\text{B’s travel time - A’s travel time}}{\text{B’s travel time}}$$

FR1 represents how many routes suggested by method A are faster than that of baseline method B, and FR2 reflects to what extent the routes suggested by A are faster than the baseline’s. Meanwhile, we use SR to represent the ratio of method A’s routes being equivalent to the baseline’s.

### C. Evaluation in the Field

We conduct two types of in the field studies: 1) The same driver traverses the routes suggested by our method and a baseline at different times. 2) Two drivers (with similar custom factors learned by our system) travel different routes simultaneously.

<table>
<thead>
<tr>
<th>Method</th>
<th>Evaluation 1</th>
<th>Evaluation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distance</td>
<td>Duration</td>
</tr>
<tr>
<td>Our System</td>
<td>15 km</td>
<td>25 mins</td>
</tr>
<tr>
<td>Gmaps</td>
<td>17 km</td>
<td>31 mins</td>
</tr>
</tbody>
</table>

Table I show the results of the two types in-the-field evaluations, where 50 users participated in the Evaluation 1 which last for 10 days and two users are invited to conduct the Evaluation 2 for six days. According to the results, 85 percent of the routes provided by our system are better than the baseline with respect to the travel time in the Evaluation 1. On average, we save 15 percent time in the Evaluation 2 (T-test: $p < 0.004$) for a 25 min trip.

### D. Evaluation By Analysis

The proposed route network evaluate with dynamic trajectory data framework in terms of both indexing overhead and storage performance. We applied Route Net on sample road networks, namely, dynamic route map and the final set of experiments.
At node 1, the speed varies at every time, then the fluctuation at node 1 speed fluctuation by speed flow analysis from this information route are analyzed and sent to the user along with traffic information.

**VII. CONCLUSION**

This system is to find out the practically fastest route for a particular user at a given departure time. Particularly the system mines the intelligence of experienced drivers from a large number of taxi trajectories and provides the end user with a smart route, which incorporates the physical feature of a route, the time-dependent traffic flow as well as the users driving behaviors (of both the fleet drivers and of the end user for whom the route is being computed). We build a real system with real-world GPS trajectories generated by over 33,000 taxis in a period of three months, then evaluate the system with extensive experiments and in-the-field evaluations. The results of our method significantly better outperforms than the competing methods in the aspects of effectiveness and efficiency in finding the practically fastest routes. Overall, more than 80 percent of our routes are faster.
than that of the existing online map services. In this extension By identifying the vehicle type, the time efficiency will improve.

REFERENCES


