Enhancing Dynamic Directions and Trajectories

C. Suganthini[1], S. Aishwarya[2], M. Shireen Sithara[3]

[1] Assistant Professor, [2] Students, Dept of Computer Science and Engineering

[1, 2, 3,] Velammal Institute of Technology, Anna University, Chennai, India

Abstract---This paper presents a smart driving direction system leveraging the intelligence of experienced drivers. In this system, GPS-equipped taxis are employed as mobile sensors probing the traffic rhythm of a city and taxi drivers' intelligence in choosing driving directions in the physical world. We propose a time-dependent landmark graph to model the dynamic traffic pattern as well as the intelligence of experienced drivers so as to provide a user with the practically fastest route to a given destination at a given departure time. 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 and customized route for end users. We build our system based on a real-world trajectory data set generated by over 33,000 taxis in a period of three months, and evaluate the system by conducting both synthetic experiments and in-the-field evaluations. As a result, 60-70 percent of the routes suggested by our method are faster than the competing methods, and 20 percent of the routes share the same results. On average, 50 percent of our routes are at least 20 percent faster than the competing approaches.

Key wods: Spatial database, data mining, GPS trajectory, driving directions, driving behaviour, GIS trajectories, route maps.

I. 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). Therefore, this service is important for both end users and governments aiming to ease traffic problems and protect environment.

A cloud-based cyber-physical system for computing practically fast routes for a particular user, using a large number of GPS-equipped taxis was proposed. In existing system, the physical feature of a route, such as distance, capacity (lanes), and the number of traffic lights as well as direction turns .The time-dependent traffic flow on the route.

A user's driving behavior. Time-dependent landmark graph, which well models the intelligence of taxi drivers based on the taxi trajectories. Disadvantages of the system are 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. 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. . In this paper, we propose a cloud-based cyber-physical system for computing practically fast routes for a particular user, using a large number of GPS-equipped taxis and the user's GPS-enabled phone.

II. RELATED WORKS

Since this paper is an extension of our previous publication we summarize the contributions (including that of the previous paper work) as follows,

In the previous paper, 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 this extension work: We further improve our routing service by self adaptively learning the driving behaviours of both the taxi drivers and the end users so as to provide personalized routes to the users. We present smoothing algorithms for removing the roundabout part of the original rough routes. We build the improved system by using a real world trajectory data set generated by 33; 000þ 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 method.

III. SYSTEM OVERVIEW

A. Intelligence modelling - 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. The suggested routes are recorded for the further use of the divers.

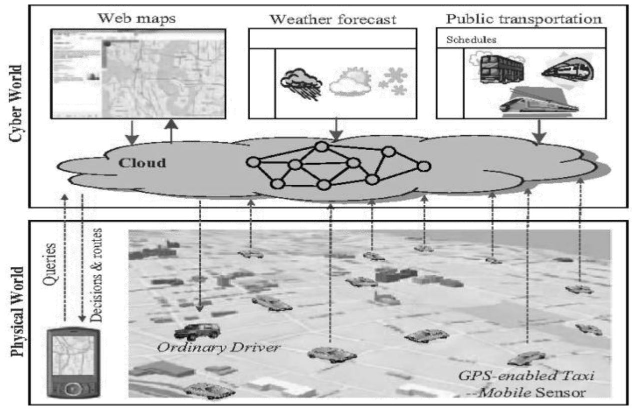


Fig.1: System Overview

B. *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 there could exists four possible routes.

Fig. 2: Low Sampling Rate Problem

C. *Route Generation* The traffic condition of a road, the travel time of a route also depends on drivers. Sometimes, different driver stake different amounts of time to traverse the same route at the same time slot. The reasons lie in a driver's driving habit, skills and familiarity of routes.

For example, people familiar with a route can usually pass the route faster than a newcomer. Also, even on the same path, cautious people will likely drive relatively slower than those preferring to drive very fast and aggressively.

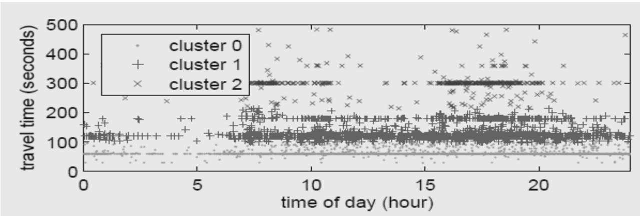
D. *Path logging* the cloud sends the computed driving routes along with the travel time distributions of the Landmark edges contained in the driving route to the phone. Later, the mobile phone logs the user's driving path with a GPS trajectory, which will be used for recalculate the user's custom factor.

The more a driver uses this system, the deeper this system under stands the driver; hence, a better driving direction services can be provided.

E. *Route computing* According to the departure time, start and destination point, the cloud chooses a proper landmark graph considering the weather information and whether it's a holiday or a workday. Based on the landmark graph, a two-stage routing algorithm is performed to obtain a time-dependent fastest route is the main output for computing the routes. These steps are followed in order to compute the practically fastest route that is very necessary for the drivers.

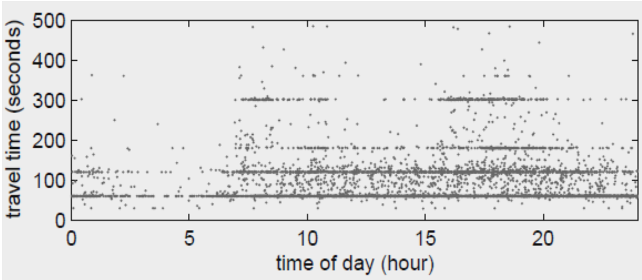
IV. TWO STAGE ROUTING ALGORITHM

A. *V clustering* We first sort T uv according to the values of travel time , 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,



(b) V-Clustering result

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



(a) Transitions of a landmark Edge

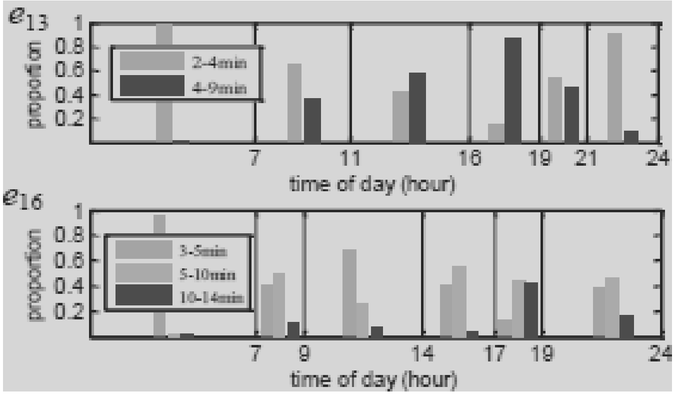


Fig.3: V - CLUSTERING

B. *E-clustering* this step aims to split the x-axis in to 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 yi with the category it pertains to and then sort the pair collection according to xi (arriving time). The information entropy of the collection Sxc is given

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

V. GRAPH CONSTRUCTION

The preprocessed taxi trajectories; we detect the top-k frequently traversed road segments, which are termed as landmarks. The reason why we use "landmark" to model the taxi drivers' intelligence is that: first, the sparseness and low-sampling rate of the taxi trajectories do not support us to directly calculate the travel time for each road segment while we can estimate the travelling time between two landmarks (which have been frequently traversed by taxis).Second, the notion of landmarks follows the natural thinking pattern of people.

The threshold _ is used to eliminate the edges seldom traversed by taxis, as the fewer taxis that pass two landmarks, the lower accuracy of the estimated travel time(between the two landmarks) could be. Additionally, we set the tmax value to remove the landmark edges having a very long travel time. Due to the low-sampling rate problem, sometimes, a taxi may consecutively traverse three land marks while no point is recorded when passing the middle (second) one. This will result in that the travel time between the first and third landmark is very long. Such kinds of edges would not only increase the space complexity of a landmark graph but also bring inaccuracy to the travel time estimation.

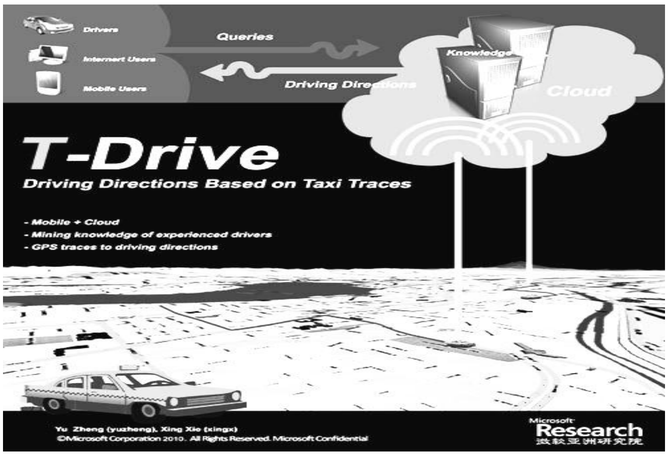


Fig.4: Working Of the System

They almost share similar traffic patterns while the weekdays and week end shave different patterns. Therefore, we build two different landmark graphs for weekdays and weekends, respectively. The travel times of transitions pertaining to a land mark edge clearly gather around some values (like a set of clusters) rather than a single value or a typical Gaussian distribution, as many people expected. This aims to handle the situation that a taxi was stuck in a traffic jam or waiting at a traffic light where multiple points may be recorded on the same road segment 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.

VI. ROUTE COMPUTING

This section introduces the routing algorithm, which consists of two stages: rough routing in the landmark graph and refined routing in the real road network.

A. *Rough Routing*

1) *Rough Route Generation:* Besides 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. The reasons lie in a driver's driving habit, skills and familiarity of routes. For example, people familiar with a route can usually pass the route faster than a newcomer. 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 custom factor as follows: Custom Factor-The custom factor _ 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. For example, _ ¼ 0:7 means that you can outperform 70 percent taxi drivers in terms of travel time under the same external conditions (traffic flow, signal, weather, etc.). Initially, we set a default value for different users. Later, in we will detail our approach for learning the custom factor for each user in a self-adaptive way with the continuous use of our service and providing a personalized route for different users. Given a user's custom factor _, we can determine his/ her time cost for traversing a landmark edge e in each time slot based on the learned travel time distribution. For example, Fig. 5a depicts the travel time distribution of a landmark edge in a given time slot (c1-c5 denotes five categories of travel times).

Then, we convert this distribution into a cumulative frequency distribution function and fit a continuous cumulative frequency curve shown in. Note 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 _ ¼ 0:7, we can find out the corresponding travel time is 272 seconds, while if we set _ ¼ 0:3 the travel time becomes 197 seconds .Now, the rough routing problem becomes the typical time-dependent fastest path (TDFF) problem. The complexity of solving this problem depends on whether the network satisfies the "FIFO" (first in, first out) property.

2) *Rough Route Smoothing:* Even using the state-of-the-art map-matching algorithm, the accuracy is less than 70 percent [4] for the low sampling rate trajectories. For example, 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. Let the rough route computed based on S, where each li is a landmark (1; 2; . . . ; n).We present a post processing to smooth the roundabout rough route. We summarize three key characteristics of a non-roundabout route, termed as non-round about Principles.

B. *Refined routing:* Suppose after the smoothing, we get a rough route. This stage finds in the real road network a detailed fastest route that sequentially passes the landmarks of a rough route by dynamic programming.

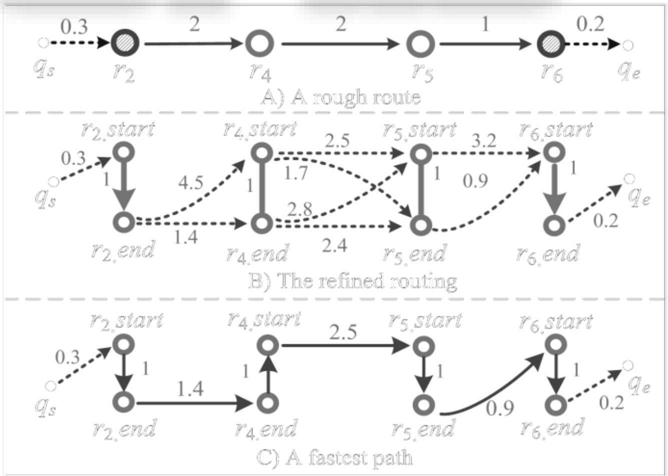


Fig.5: Refined Routing

Assume r1; r2; . . . ; rn are the corresponding road segments .each has its start point and end point be the earliest leaving times (after traversing ri) at nodes and , respectively. Let be the travel time of the fastest route from road nodes a to b without crossing node c. Let the time (estimated based on speed constraint) for travelling from node a to b.

C. *Learning Custom Factor*

This section describes the process for learning the user's custom factor and providing self-adapted fastest route, which contains five steps as follows,

1) *Query sending.* First, the user sends her query tuple to the cloud, where qs and qd are start point and destination and td is the departure time.

2) *Route computing.* According to the departure time, start and destination point, the cloud chooses a proper landmark graph considering the weather information and whether it's a holiday or a workday. Based on the landmark graph, a two-stage routing algorithm is performed to obtain a time-dependent fastest route.

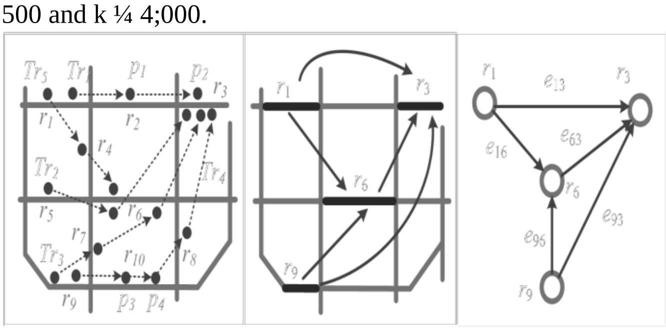
3) *Path logging.* The cloud sends the computed driving routes along with the travel time distributions of the landmark edges contained in the driving route to the phone. Later, the mobile phone logs the user's driving path with a GPS trajectory, which will be used for recalculate the user's custom factor. The more a driver uses this system, the deeper this system understands the driver; hence, a better driving direction services can be provided.

4) *Adapting the custom factor*. The custom factor of a given user can be learned in a self-adaptive way. Initially, we assign the user a default value, be the custom factor the client sent to the cloud for the Mth query. Let cumulative distribution function for the ith landmark edge. After the travel, we calculate the real travel time of this landmark edge i by the recorded GPS logs. Then, the mobile client computes the new custom factor by where p is the number of landmark edges. This single valued minimization problem can be solved using the optimization approaches or just using the simple enumeration method (uniformly trying the from 0 to 1. To obtain a stable value for, we need to study the most recent n driving routes of a user instead of a single trip. Meanwhile, near past driving paths should be more valuable in calculating than those distant past. Therefore, we compute the new personalized by a weighted moving average where n is the window length of the moving average. In the next query, the updated result will be sent to the cloud.

VII. EVALUATION ON TIME TRAVEL ESTIMATION

A. *Evaluating Landmark Graphs*

We build a set of landmark graphs with different values of k ranging from 500 to 13,000. The threshold _ is set to 10i.e., at least 10 times per day traversed by taxis (in total over 900 times in a period of three months) and tmax is set to 30 minutes Visualizes two landmark graphs when k ¼ 500 and k ¼ 4;000.



A) Matched taxi trajectories B) Detected landmarks C) A landmark graph

Fig.6: Evaluating Landmark Graph

The red points represent landmarks and blue lines denote landmark edges. Generally, the graph well covers Beijing city, and its distribution follows our common sense knowledge. There by next comes the evaluation on routing. This is also processed similar to that of the evaluation of the land mark graph.

B. *Evaluation on Routing*

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**.**

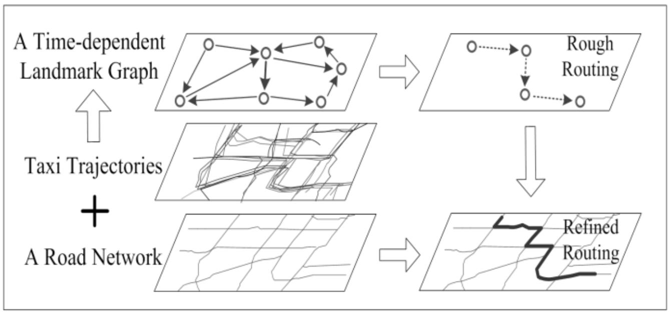


Fig.7: Trajectories and Routing

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 an 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.

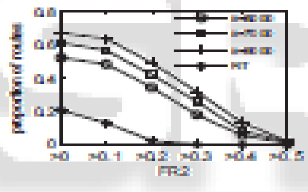


Fig. 15. FR2 over *k*

TABLE 2
FR1, SR of TDrive and RT

	α	k	FR1	SR
TDrive	0.4	6,000	0.509	0.281
	0.4	9,000	0.647	0.222
	0.6	6,000	0.511	0.272
	0.6	9,000	0.653	0.216
	0.7	6,000	0.544	0.227
	0.7	9,000	0.672	0.214
RT approach			0.206	0.671

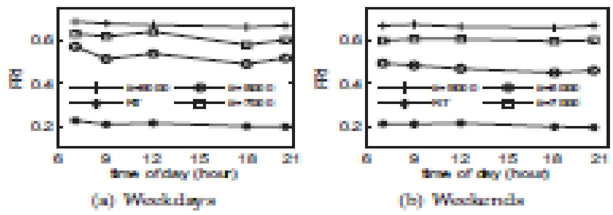


Fig. 16. FR1 w.r.t. time of day

Fig.8: TRAFFIC PATTERN WITH RESPECT TO A DAY

The reason why our method outperforms the RT approach is: 1) Coverage: Many road segments have neither embedded road sensors nor taxis travelling on them at a given time. At this moment, the speed constraint of a road segment is used to represent the real-time traffic on the road segment. That is also the reason why the RT approach returns many of the same routes as the SC method. 2) Spareness: Usually, we cannot have enough number of the taxis travelling on a road segment in a near past time interval, e.g., past 5 minutes. Thus, the instant travel time (so called real-time speed) estimated based on these insufficient samples is not very accurate. 3) Open challenges: As compared to the history based method, the RT approach is more vulnerable to noise, such as traffic lights, human factors (pedestrians crossing a street), and taxis looking for parking places and passengers.

VIII. PERFORMANCE AND IMPLEMENTATION

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 (recommended by different methods) simultaneously. Table 1 show the results of the two types in-the-field evaluations, where 30 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, 79.4 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 15.5 percent time in the Evaluation 2 (T-test: $p < 0{:}004$) for a 25 min trip.

Table 1: Trajectories of the In-the-field Study

	Evaluation 1	Evaluation 2
Num. Trajectories	360	60
Num. Users	30	2
Total Distance (km)	5304	814
Total Duration (hour)	165.24	25.09
Evaluation Days	10	6

Table 5: In-the-field Evaluation 1

	T-Drive	Google	Δ	R1	R2
Distance	13.91km	15.56km	1.65km	0.517	0.106
Duration	25.80min	29.28min	3.48min	0.808	0.119

Table 6: In-the-field Evaluation 2

	T-Drive	Google	Δ	R1	R2
Distance	13.58km	13.55km	-0.03km	0.367	-0.002
Duration	23.18min	27.00min	3.82min	0.750	0.141
WaitTime	4.77min	6.50min	1.73min	0.633	0.267

Fig.9: Implementation and Analysis

The traffic pattern and trajectories may vary according to the week end and week days which can be shown clearly as above. Our system is said to give the maximum performance when compared to that of the other mapping functions.

IX. CONCLUSION

Thus it describes a system 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 incorporate the physical feature of a route, the time-dependent traffic flow as well as the users' driving behaviours (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, and then evaluate the system with extensive experiments and in-the-field evaluations. The results show that our method significantly outperforms the competing methods in the aspects of effectiveness and efficiency in finding the practically fastest routes.

REFERENCE

[1] J. Yuan, Y. Zheng, C. Zhang, W. Xie, G. Sun, H. Yan, And X. Xie, "T-Drive: Driving Directions Based On Taxi Trajectories," 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] A. Orda And R. Rom, "Shortest-Path and Minimum-Delay Algorithms In Networks With Time-Dependent Edge-Length," J. Acm, Vol. 37, No. 3, Pp. 607-625, 1990.
[7] N. Leibowitz, B. Baum, G. Enden, and A. Karniel, "The Exponential Learning Equation As A Function Of Successful Trials Results In Sigmoid Performance," J. Math. Psychology, Vol. 54, No. 3, Pp. 338-340, 2010.
[8] B.C. Dean, "Continuous-Time Dynamic Shortest Path Algorithms," Master's Thesis, Mit, 1999.
[9] C. Schensted, "Longest Increasing And Decreasing Subsequences," Canadian J. Math., Vol. 13, Pp. 179-191, 1961.
[10] W.S. George and G.C. William, Statistical Methods, Eighth Ed. Wiley-Blackwell, 1991.
[11] 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.
[12] C.Defabritiis, 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. 200