ROAD TRAFFIC CONGESTION ESTIMATION SYSTEM USING GPS DATA

[1]HNIN THANT LWIN, [2]THINN THU NAING

[1]University of Computer Studies (Yangon), [2]University of Computer Studies (Yangon)
E-mail: hninthantlwin@gmail.com,thinnthu@gmail.com

Abstract- Traffic congestion is a major headache problem in urban area in both developing and developed countries. Congested roads can be avoided by determining the travel-time for a particular road ahead of time. In this paper, we present the traffic congestion estimation system using hidden markov model. We use data from GPS-enabled phones on vehicles to estimate the traffic conditions for cheaper and real time travel-time estimation. We accept the user's query including GPS data from the mobile phone and communicate with the cloud and present result to user. When we get the real time GPS data from mobile phone in vehicle, we use Map Matching algorithm to match these GPS data to road network to know which vehicles are on which roads. Framework based on Hidden Markov model is used to estimate the traffic congestion on each road by considering both historical traffic patterns and present traffic flow data. Then, we can choose the less congested route from these estimation results.

Keywords- Traffic Estimation, GPS, Hidden Markov Model.

I. INTRODUCTION

Urban traffic congestion is a severe problem that significantly reduces the quality of life in particularly metropolitan areas. However, frequently constructing new roads is not realistic and untenable in social and economic aspects. Therefore, the urban traffic management such as traffic congestion estimation systems has been conducted by many researchers. Traditionally, estimating travel times has relied on slow and costly methods such as loop detectors, observations vehicles or automatic vehicle identification or floating car observers, car GPS. However, the increasing popularity of mobile phones embedded with positioning functionality such as GPS is allowing users to easily acquire their own locations and collect their own trajectories, which can be used for various purposes such as location-based service applications. Therefore, new possibilities have opened for cheaper travel-time prediction.

Although dedicated moving observer or floating car vehicle-based methods can provide precise estimations, they require that an instructed driver collected the data needed. This is both time consuming and costly as the driver must be paid. This method also provides less data as a relatively small number of vehicles are usually used. Since road networks are ever changing and traffic volumes fluctuate, travel-time estimates must be recalculated occasionally or continually using current data to reflect these changes. For this reason, we use GPS data from mobile for the efficient and effective methods.

Previous approaches to travel-time estimation include algorithms based solely on more or less educated guesses calculated from the permitted speed on a particular road segment, on finding weighted average given single observations, on data collected using expensive moving observer methods, or on the experience of traffic experts. There are many approaches for estimation traffic in urban areas such as data fusion, fuzzy control theory and microscopic traffic simulation, etc; but all these techniques need costly traffic data.

In this paper, we collect data from mobile GPS on the vehicles. We then map these GPS data with the road network to know which vehicles are on which road. We then use Hidden Markov model to estimate the traffic on a particular road ahead of time using these GPS data and historical data. This paper intends (1) to estimate travel time of a route for a vehicle driver to choose fastest path to the target destination, (2) to minimize travel time and cost by avoiding traffic congestion that leads to be wastes a lot of gas and fuel (3) to calculate the traffic data from both historical and recent dynamic traffic data and to send the accurate result to mobile users.

II. RELATED WORD

Several papers are concerned with prediction travel-time estimation and actual path finding. Kanolus et al. propose a method for finding the fastest path through a road network given the constraints of a time interval at either the start of or destination of the trip. Ku et al. propose an adaptive nearest- neighbour query based on travel time instead of Euclidian or network distance.

In Ludger Hovestad, Vahid Mossavi proposed a conceptual data driven traffic modeling framework, which is mainly based on the application of Markov chain in a continuous coexistence with data stream from GPS data on taxi cabs. In GPS equipped vehicles are used to collect samples at regular intervals, which are then used for estimating travel times. Many

different sampling intervals are used 30 seconds in and one second in. Different systems will provide data recorded with different sampling rates, a solution independent of sampling rates has not been proposed to our knowledge. In Quiroga et al. study the impact of changing sampling rate and road segment length using GPS.

Yan Qi presented probabilistic models for short term traffic conditions predictions and compared the traffic prediction using HMM based model and one step stochastic model. He derived traffic features from embedded magnetic loop in the road.

In Yannis. George, Constantinos Antoniou and Hoaris N. Koutsopoulos describe a methodology for the identification and short-term prediction of traffic state. This methodology comprises the components such as model-based clustering, variable length Markov chains and nearest neighbor classification.

In Jing Yuan, Yu Zheng, Xing Xie, Guangzhon presents a Cloud-based system computing customized and practically fast driving routes for an end user using historical and real time traffic conditions and driver behaviour. The cloud builds a model incorporating day of the week, time of day, weather conditions and individual driving strategies. This paper infer the future traffic conditions on a road using an m^{th}-order Markov model and this condition is integrated into the proposed routing service.

III. OVERALL ARCHITECTURE OF THE TRAFFIC ESTIMATION

The general architecture for traffic prediction is described in following figure.

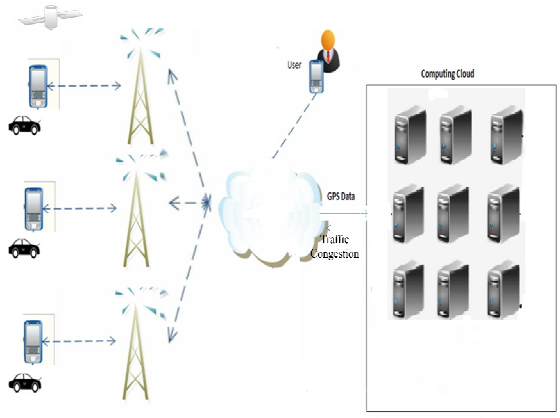


Figure 1: General architecture for Traffic Prediction.

Android device on vehicles is used for capturing GPS location data, source and destination and transmitting the data to data center. GPS data provide with longitude, latitude, a timestamp, direction, speed and a unique identification for the recording GPS device. The data transmitted by android devices deployed in vehicles is collected and in a computing cloud. These GPS data from mobile phone on vehicle is matched with the road network using map matching algorithm to know vehicle's location. We use decision rule map matching algorithm to do this. Traffic conditions are estimated using Hidden Markov model using these GPS data and historical data. To carry out these processes, we used cloud computing as back end server. Then, we display the estimated traffic congestion results to the user's mobile phone.

IV. PROCESS FLOW OF THE SYSTEM

The processes of backend sever in cloud computing to estimate traffic congestion condition is described in following figure.

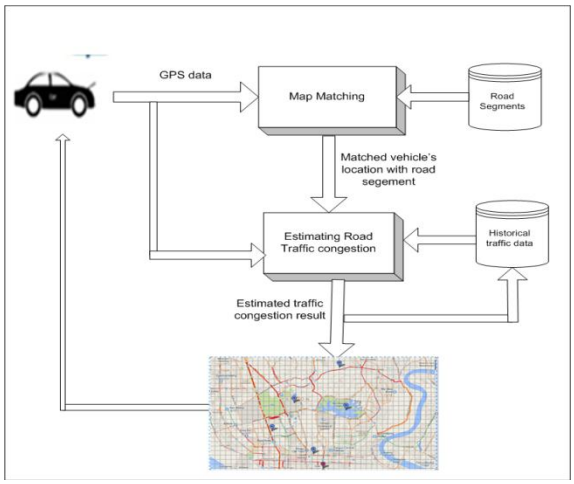


Figure 2: Process flow of the System

Detail process of estimating traffic congestion system of this paper is presented in the followings:

A. Road Segment

A road segment r is a directed (one way or bidirectional) edge that is associated with two terminal points start point (r.s) ,end point (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 start point r.s to end point r.e , otherwise, people can start from both terminal points, i.e, r.s → r.e or r.e → r.s. To create road segment, we gather road networks of Yangon from Google Map and to get the latitude and longitude of each road segment we collect data in the open street map data. Then, we store the road segment data to the database with corresponding additional information for each road segment.

g_id	segment_name	traffic_lights	shape_file
1	hledan_sanyeiknyein	FALSE	0105000000100000102000080000092AD2EA704D4340CD902ABA406C3F1
2	hledan_marlar	FALSE	0105000000100000102000074000041E261D422C213042A801F590860E2CF1
3	sanyeiknyein_sinyaytwin	TRUE	0105000000100000102000001F0000E960FD9FC3D4304068924EFD4B085844
4	marlar_tadarphyu	TRUE	0105000000100000102000A30000B502D86481AC430758240FA3A143295
5	sinyaytwin_butaryoune	FALSE	0105000000100000102000024000076E09C11AD53040925A02d23908540E2
6	tadarphyu_kyakwae	FALSE	0105000000100000102000002C000036591A2449301482794 1E2A603572D53
7	butaryone_thuka	FALSE	0105000000100000102000D0000B579C3657D630402B81A5962C08584076
8	thuka_thanlan	FALSE	0105000000100000102000 1B0000A3F161411CD73040778A7D35210858408
9	bartar_okkin	FALSE	0105000000100000102000012000022CEBDE2CDD930404D6688E0F90758407
10	okkin_phayarlan	TRUE	0105000000100000102000480000E39283410BA74061579B1D37311558F453
11	hledan_sitepyoyay	TRUE	0105000000100000102000009100087 1E53A816C53601DA8041B9C5074C2E

Figure 4: some portion of road segments database

B. Matching GPS Data with Road Segments

The increasing popularity of GPS-enabled device has facilitated users to track moving objects such as vehicles and people. However, as the readings of a GPS sensor have positioning errors and sampling errors, the departure of the GPS tracking data from the actual trajectory can hardly be avoided. Map matching is the process of matching a raw GPS tracking data to road network. To get the real time estimation of traffic congestion, we use the GPS data collected from the mobile phone on vehicles. GPS data for a vehicle trajectory (Tr) consists of a sequence of GPS points pertaining to one trip. Each point p consists of a longitude, latitude, speed, timestamp and a unique identification for the regarding GPS device.

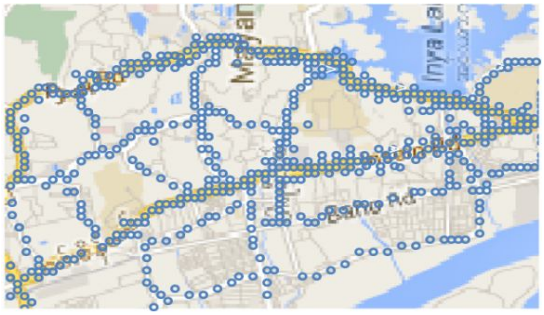


Figure 3: Some GPS points from android device on vehicles

To know which vehicles are on which road, we map these GPS data with the road segments. However, there is a problem of correctly matching a sequence of GPS sampling points to the roads on a digital map. To solve this ambiguity, we use decision-rule topological map-matching algorithm. It determines the correct roadway centerline for vehicle travel by obtaining feasible shortest paths between snapped GPS data points in post-processing mode. The algorithm selects all roadways within a buffer around a GPS data point and snaps the point to the closest roadway by obtaining the minimum perpendicular distance from the data point to each roadway. To calculate the closest roadway with the GPS point of the vehicle we use the Spherical Law of Cosine formula because it gives well-conditioned results down to distances as small as a few meters on the Earth's surface. To reduce the calculation time we first load the road way from the database within 50 meters with the GPS point and then snap the GPS point (p1) to the closet roadway within the buffer.

Input: set of road segments(r_s), GPS trajectories (k_0,….k_j)
Output: snapped GPS point

1. for each road segments in r_i in r_s
   getBoundingBox(k_i.lat, k_i.lang,distance)//
   If((minlat< r_i.lat <maxlat)
   and (minlang< r_i.lang<maxlang))
   bf[]ć r_i
   end if
   end for
2. for each roadsegment r_i in bf[]
   Calculate_distance(k_i.lat,k_i.lang,r_i.lat, r_i.lang)
   r_{min} ć min(distance)
3. snap point (k_i) to r_{min}

Figure 5: Algorithm for snap GPS points with the road network

When we finish snapping this GPS point with road network we determine whether this vehicle is exist in this road using the next point. We calculate the shortest path between this snapped point and the newly-snapped GPS data point (p2). If the path between these two points is not feasible, then we determine if feasible routes exist between the preceding and subsequent points bounding the GPS data points of concern. Therefore, we look ahead by snapping next newly-snapped GPS data point (p3) to nearest roadway centerline within its buffer and determine if the shortest path between snapped points (p2) and (p3) is possible. If the tested path is not feasible, we snaps point (p2) to the next nearest roadway centerline within its buffer around point 3 that have not already been used in a feasibility path check. The following diagram show the steps of the map matching algorithm that use in this paper.

Input: set of road segments(r_s), GPS trajectories (k_0,….k_j)
Output: map matching results

1. snap point (k_i) to r_{min}
2. if k_i and k_j are in same road r_{min}
   Match points with r_{min}
   End
3. Else if k_j and k_{j+1} are in same road
4. If k_i has another nearest road segment in buffer[]
   Snap point k_i to another r_{min1}
   K_sć k_i
   If k_{s-1}, k_s, k_{s+1} are in same r_{min1}
   Match points with r_{min1}
   End
   Else
   Mismatch point
   End
   End if
5. Else if k_{i-1} and k_i are in r_{min1}
   Match points with r_{min1}
   End
   Elseif find k_{i-1} has another nearest road segment in buffer[]
   Snap k_{i-1} to another r_{min2}
   K_sć k_{i-1}
   If k_{s-1}, k_s, k_{s+1} are in same r_{min2}
   Match points with r_{min2}
   End
   Else If k_{i-2} and k_j are in same road r_{min3}
   Match point with r_{min3}
   End

Else
Mismatch point with r_{min3}
End
End If
End If
6. Else if k_j has another nearest road segment in bf[]
Snap the k_j to r_{min4}
K_sć k_j
If k_{s-1}, k_s, k_{s+1} are in same r_{min4}
Match points with r_{min4}
End
Else
Mismatch point
End
End if
7. Else if k_i and k_{i+1} are in same road r_{min}
Match point with r_{min}
End
Elseif k_i has another nearest road segment in buffer[]
Snap point k_i to another r_{min1}
K_sć k_i
If k_{s-1}, k_s, k_{s+1} are in same r_{min1}
Match points with r_{min1}
End
Else
Mismatch point
End
End if
End If
End If

Figure 6:-setp sequence of Matching Algorithm

In the following figure 7 describes the result of map matching of a road segment in our system.

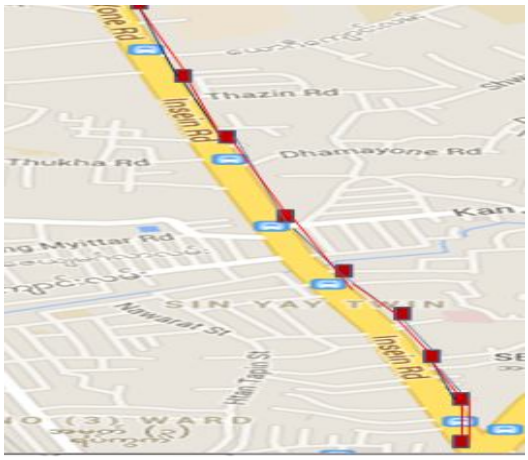


Figure 7: A screen shoot of a map matching result

C. Estimation Traffic

When we have the map matching result of vehicle trajectories data with road segment and we know the vehicle's location, we estimate the traffic congestion conditions. To estimate the real time traffic condition at a time (T) on each road network use both historical data on each road network (H) and the recent GPS trajectories data of mobile phone on vehicle (R).

To get the historical data for each road segment on each time, we collect the data based on weekends and weekdays. We collect the traffic congestion probability of each road segment in each minute of time window and marked which road segment is the most traffic congestion area in the city and traffic lights exist on which road segment and how long it is. For real time data, we can get it from the GPS enabled phone from the vehicle.

There are many available models to estimate the traffic and these models can be categorized based on their way of computation to different kinds of statistical models, simulation models and artificial intelligence models. In this paper, we use the Hidden Markov model to estimate the traffic congestion of time (T). It is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. In a regular model, the state is directly visible to the observer and therefore the state transition probabities are only parameter. In a hidden Markov model, the state is not directly visible, but output, dependent on the state is visible.

Markov chains have some other interesting features that can be used for specific task such as finding critical urban segments, empirical expected travel times, community detection, road engineering and traffic management. Traffic conditions are considered as hidden states and traffic parameters observations are symbols. It matches the basic structure of HMMs, and therefore, an HMM is suitable for traffic modeling.

Hidden Markov Model is represented by $\lambda = (A, B, \pi)$, N , the number of states in the model, M, the number of distinct observation symbols.

- The transition matrix for state distribution is intialized randomly and is represented by A
- The initial state of matrix of observation probabilities is represented by B and
- A vector of initial probabilies is represented by π.

We define three states: traffic jam, traffic heavy and traffic smooth and three observation symbols: speed, traffic jam zone and peak hour. We also have to define the transition probability matrix (A) and observation probability (B) and start state (π). To define the transition probability matrix we use the Baye's theorem.

$$P(x|y) = \frac{P(y|x).P(x)}{P(y)} \qquad (1)$$

Where x is the current states of the traffic congestion (traffic jam, traffic heavy, traffic smooth) and y is the historical states of the traffic congestion states of each road segment in time (t). $P(x|y)$is the probability of

current states given historical state, $P(y|x)$ is the probability of historical state given current states and $P(x)$ is the probability of current state and $P(y)$is the probability of historical states. We already predefined the observation probability for traffic jam zone and traffic lights for each road segment in time (t). When we get the GPS data of vehicles in on the road network, we also get the speed of vehicles riding on the road. We define the speed as: traffic jam : if the speed is γ ; traffic heavy : if the speed is ρ ;traffic smooth : if the speed is υ; where γ, ρ and υ are threshold values.

We calculate the probability of traffic congestion state using the number of speed that can whether traffic jam, traffic heavy or traffic smooth in a time window; we count the number of speed that can occur traffic jam in a time window (one minute). Then, we can model the traffic estimation with the Hidden Markov Model with the corresponding transition probability matrix, observation matrix and start state. In the framework of HMMs the problem of traffic condition estimation can be expressed as an optimization problem. The Viterbi algorithm was used to search for the optimal states sequence, which is based on dynamic programming methods. Therefore, we use Viterbi algorithm to search the optimal states sequence of traffic estimation. From this, we get the probability of traffic congestion states (traffic jam, traffic heavy and traffic smooth). We identify traffic jam as red color, traffic heavy as yellow color and traffic smooth as green color based on this probability of these traffic congestion results. We integrate these results with the Google Map and show the traffic estimation results to the user as the following figure.

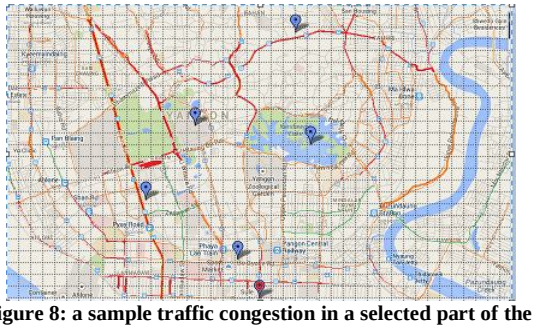


Figure 8: a sample traffic congestion in a selected part of the Yangon; Red for jam, Orange for heavy and Green for smooth traffic.

V. EVALUATION OF TRAFFIC ESTIMATION

To quantify the accuracy of the traffic inference, we use the root mean square error (RMSE) define as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}(x_i - \hat{x}_i)^2} \qquad (2)$$

Where x_i is the real travel time and $\hat{x}_i$ is the estimated travel time and N is the number of estimation. The following figure shows the RMSE of a road segment (hledan_sanyeitnyein) of real traffic and estimated traffic using real time data real time and historical data of our system.

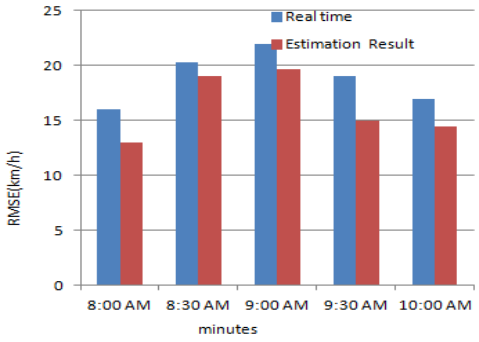


Figure10: RMSE of a road segment

CONCLUSIONS

This paper describes a system for drivers to compute shortest-time driving routes using traffic information. We use real time traffic data on mobile phone in vehicles' GPS and match these GPS data with the road segments using the decision rule map matching algorithm to know the location of vehicles. In this paper we use Hidden Markov Model to estimate traffic congestion for more accurate result using real time GPS data and historical data.

REFERENCES

[1] B.S. Yoo, S.P. Kang, and C.H. Park. Travel time estimation using mobile data. 2005.

[2] Barbara Frith, David Pearce, and Tom Sutch. The highways agency journey time database. Road Transport Information and Control, pages 98–105, 2004.

[3] Carola A. Blazquez, Chatpter 9 in "Global Navigation Satellite Systems-Signal,Theory and Application "

[4] Lasgouttes J M ,Furtlehner C and Fortelle A D L 2007 A belief-propagation approach to traffic prediction using probe vehicles Proc. 10th IEEE Conf. of Intelligent Transportation Systems pp10221027

[5] Ludger Hovestad, Vahid Mossavi ,"Modeling Urban Traffic Dynamics in Coexistene with Urban Data Stream" Augest 11,2013 , Chicago,USA

[6] Ming-Hui Chen, Qi-Man Shao, Joseph G. Ibrahim, Monte CarloMethods in Bayesian Computation, Springer, New York, 2000.

[7] Michael A.P. Taylor, Jeremy E. Woolley, and Rocco Zito. Integration of the global positioning system and geographical information systems for traffic congestion studies. Transportation research. Part C : Emerging technologies, pages 257–285, 2000.

[8] Nectaria Tryfona, Dieter Pfoser, and Agnes Voisard. Dynamic travel time maps - enabling efficient navigation. In SSDBM '06: Proceedings of the 18th International Conference on Scientific and Statistical Database Management (SSDBM'06), pages 369–378, 2006.

[9] N. Quiroga Cesar A and Darcy Bullock. Travel time studies with global positioning and geographic information systems

[10] Rajput D., Singh P., and Bhattacharya M., "An Efficient and Generic Hybrid Framework for High Dimensional Data Clustering," 2010.

[11] Tian Xia, Evangelos Kanoulas, Yang Du, and Donghui Zhang. Finding fastest paths on a road network with speed patterns. In ICDE '06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), pages 1–10, 2006.

[12] Wei-Shinn Ku, Roger Zimmermann, Haojun Wang, and Chi-Ngai Wan. Adaptive nearest neighbor queries in travel time networks. In GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems (GIS'05), pages 210–219, 2005.

[13] X.David Doria," Expectation-Maximization: Application to Gaussian Mixture Model Parameter Estimation", April 23, 2009

[14] Xing Yi, Baibo Zhang, Changshui Zhang, Competitive EM Algorithm for Finite Mixture Models, Pattern Recognition, Volume: 37, Issue: 1, January, 2004,

[15] Xing Xie, Jing Yuan, Yu Zheng, Guangzhon, "Driving with knowledge from physical world", Feb, 2013

[16] Yannis. George, Constantinos Antoniou and Haris N. Koutsopoulos "Traffic state prediction using Markov chain Model", Julu , 2007

[17] Yan Qi ".probabilistic models for short term traffic conditions",2010.

★★★