International Journal of Technology and Engineering System (IJTES)
Vol 7. No.4 2015 Pp. 402-405
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

CLASSIFICATION OF TRAJECTORIES IN LIVE TRAFFIC USING VARIANCE BASED APPROACH

S.MUTHUSELVI[1], R.BHUVANYA[2]
[1]PG Student, [2]Assistant Professor.
Department Of CSE, K.Ramakrishnan College Of Technology, Samayapuram, Trichy.
bhuvanyaraghunatham@gmail.com, sugumar.selvi92@gmail.com

ABSTRACT

To mine a smart navigation system from a wide range of real-world historical trajectories for practically suggests the fastest route to a user according to his\her departure time. The trajectories are aggregated and mined in the cloud database. As per Live Traffic, the trajectories are constantly updates in the cloud database. This Live Traffic Index (LTI) is used to the intelligence navigation system to analyze traffic circumstances and reduce the online computation of route identification. LTI construction is designed to learn the time-variant distributions of the travel times between any two landmarks. Besides the traffic flow, our method also incorporates additional factors such as pattern recognition, in which this pattern can intimate the user before he\she enters a particular location. Here extend our framework to analyze weather patterns to implementing variance entropy based clustering algorithm to improve the efficiency and calculating speed results to update the traffic routes.

Keywords: Trajectories, Live Traffic Index, Pattern recognition, Traffic flow, shortest path.

I .INTRODUCTION

The growing popularity of online mapping services such as Google Maps and Microsoft MapPoint has led to an interest in responding areal time queries such as finding shortest routes between locations along a spatial network [11] as well as finding nearest objects from a set *S* (e.g., gas stations, markets, and restaurants). Elements of *S* are usually constrained to lie on the network or at the minimum to be easily accessible from the network. Finding efficient navigation system has become a daily activity and been implemented as a key feature[13] in many map services like Google, Bing Maps and Keyword search over a large amount of data is an important operation in a wide range of spatial domains. Implemented raw transmission model to analyze the traffic data with time stamps. Dijkstra's algorithm [2] which is known as uniformed search approach to find shortest path. A few updates could affect a large portion of packets. The response time is relatively high and the clients may receive large amount of irrelevant updates due to the transmission model. In existing system the interaction between users and search was limited. As search engines [5] were not personalized so privacy is not maintained. It does not differentiate content and location concept so it effect on performance of system. User profiles are not properly maintained. All processing tasks are done on the client side so it may decrease the performance. A search engine does not think personally. Approximate string search could be necessary when users have a fuzzy search condition or simply a spelling error when submitting the query, or the strings in the database contain some degree of uncertainty or error. In the context of spatial databases approximate string search could be combined with any type of spatial queries, including range and nearest neighbor queries [12]. 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. 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 Dijkstra's algorithm have been combined as hierarchal and Goal-Directed Speed-Up techniques to Speed up the query processing time[2].But this technique is suitable only for high level of hierarchies. This increases the uncertainty of the routes traversed by a taxi. Since the road network is dynamic, we can use neither the same nor a predefined time partition method for all the landmark edges.

II . PROPOSED SYSTEM

In early days, Dijkstra's algorithm is used to find the shortest path but it have some drawbacks such as precomputed data, irrelevant results for clients, response time relatively high etc. Typical client-server architecture can be used to answer shortest path queries on live traffic data. In this case, the navigation system typically sends the shortest path query to the service provider and waits the result back from the provider (called result transmission model). However, given the rapid growth of mobile devices and services, this model is facing scalability limitations in terms of network bandwidth and server loading. To overcome this drawback we propose the LTI (Live Traffic Index) which enables drivers to quickly and efficiently collect the live traffic information on the broadcasting channel. With the help of LTI, shortest path can be determined. In which path can be dynamically changing based on the Live Traffic Updating. Our method can be mine a smart navigation system from a large number of wide-range historical trajectories. Some of the merits of LTI, which it reduce tune-in-cost accomplished while the data are fetched from the air medium.
Scalability plays a vital role in LTI. In which specific shortest path is extract through the broadcast an air index over the wireless network. The main advantages of this model are that the network overhead is independent of the number of clients and every client only download a portion of the entire road map according to the index information.Architecture design for the system is proposed; and a plan for iterative, incremental nature of ensuring the project is developed. Figure 1 depicts the system architecture that explains each and every modules of the process.

Initially the user had given the source point (text) as input. This source point undergoes a sequence of steps such as shortest path computation, traffic circumstances and reconstructs the result. First, shortest path computation, construct the road network that contains the pre-processing data for analysing the shortest path. Next, Traffic circumstances based on the LTI construction, transmission, maintenance process. Finally, reconstructed result is analyzed based on the traffic updating and shortest path computation.

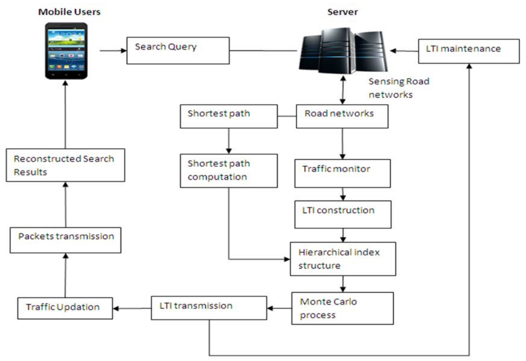

Figure 1 System Architecture

A cloud based system is proposed for computing fast routes practically for a particular user, based on the large number of GPS- equipped taxis and the mobile phones. As shown in figure 3, first GPS equipped taxis are considered as a mobile sensors searching for the traffic measures of the road in the real world. Second, cloud system is mined and aggregate the information from the taxis and other sources like web maps, weather condition. In that mined information contains the driving directions of taxi driver and traffic rhythms on the road shallow. The cloud knowledge is used for both GPS enabled

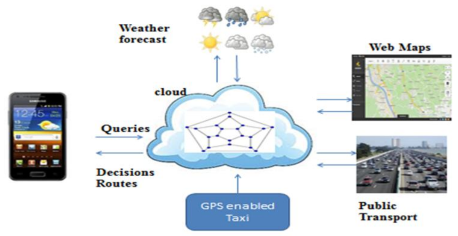

Figure 2 weather condition

users and ordinary drivers in the real word road networks. Finally the mobile client in a GPS- phone accepts the user's query and then interact with the cloud and offering the best result to the user.

III . LIVE TRAFFIC INDEX

The traffic details are updated based onthe index transmission model.so create a framework called Live Traffic Index (LTI). Figure 3which enables to collect the live traffic information from the broadcasting.

Figure 3 Traffic circumstances

Pseudo code for LTI construction

NO	STEPS
1	Get the Graph for road network.
2	Graph G as (V_g, E_g) with vertex and Edges
3	Create sub graphs $SG_{i...}$ SG_n
4	Analysis connectivity edges for each sub graph

5	Find shortest path routes with precomputation distances
6	Index creation for each sub graphs and edges
7	Graph partitioning with cheeger and cuts
8	Transmission index with shortest path routes
9	Read Head segment
10	Broadcast with sub graphs

Table 1 Pseudo code for LTI construction

1. LTI construction

This construction is based on the Hierarchical Index Structures. Monte Carlo process is used to execute a set of randomly generated shortest path queries on a temporal index. It represents the stochastic based index construction that minimizes not only the size overhead but also reduces the search space of shortest path queries.

2. LTI Transmission

In this sub module, transmission willgets broadcasting scheme to listening to the header of data packets. Analyzed format of data packets contains id and checksum values. The broadcasting model uses radio or wireless network (e.g., 3G, LTE, Mobile WiMAX) as the transmission medium. When the server broadcasts a dataset, all clients can listen to the dataset concurrently.

3. LTI maintenance

Implement an incremental update approach that can efficiently maintain the live traffic index. To keep the freshness of LTI, every sub graph is required to maintain its corresponding shortcut edges. The entire update process is done at the service provider and there is no extra data structure being broadcasted to the clients.There is a bottom-up update framework to maintain the hierarchical index structure. Theupdated sub graphs starts from lowest level to root.

4. Pattern recognition

Pattern recognition is used to find the different pattern format in the road networks. This pattern will intimate the user, before he/she enters the particular location. So easily identify the patterns while entering the target location. In addition we analyze the weather condition for source to target destination.

IV . VARIANCE BASED ENTROPY CLUSTERING ALGORITHM

In this algorithm helps to overcome the limitations of K-Means clustering algorithm. This clustering algorithm is used to analyze the weather patterns and also speed analysis in road networks. A road network is dynamically changing that can't be a pre computed one. Based on the traffic flow updations will be maintained in server side.

1. VE-Clustering

It collects the traveling time values and sort the values in partition order in recursive way. In this partition, First, find out the variance in travel times. Finally, compute the "best" point having the minimal variance time.

The variance based clustering algorithm performs in three forms.

1. It calculates the minimum points of the each element
2. It arranges the elements in Entropy format which is called as Applicant set that is inimitable and usual. It does not have any unwanted elements.
   1-12, 2-7, 3-11, 4-8, 5-9
3. Then it arranges the data in a descending order
   1, 3, 5, 4, 2
   In third form Euclidean distances are applied in clustering.

a. Design Descending data order data in the Entropy.
   This form recognize the Entropy of each element in the given data.
   S={E1, E2, E3…..E4} and place them in descending order of the element entropy
   For example,
   S={1,2,2,3,1,1,4,4,1,3,4,3,3}

Element	1	2	3	4
Entropy	4	2	4	3

   Given the data in descending order
   1:4, 3:4, 4:3, 2:2

b. Calculation of Applicant Set A from DataD
   This form decides the Applicant element in the data.
   We have 4 data , at the place of 13 data elements.
   By using clustering can decrease the number of calculations.

Cluster={1,3,4,2}.

c. Cluster Creation
   This form contains three steps,
   1. Based on the Euclidean Distance applied the clustering technique by using Applicant Set.
   2. Remaining elements are rearranged in s
   3. Remove the empty clusters.

Algorithm
Input:
E= { E1,E2,……,En} /n is a set of Data Elements
K // number of clusters
Output:
A set of K clusters.
Procedure

```
a. Evaluate the entropy based on each data element in E.
b. Entropies are arranged in descending order and call them as elements.
c. Create the Applicant Set A. In that Set there is no duplicates in cluster and create one duplicate Applicant Set EC
d. i. Fix the CLk as 0 for each cluster CC and call that as a center of cluster CC
e. ii. Assign a element for every cluster CLk from the Applicant Set A.
f. Recompute the mean CL
```

CONCLUSION

Thus the shortest path results are computed or updates based on the Live Traffic circumstances. This experiment provides the optimal path for variance performance factor for online shortest path computation. Here extend our framework to analyse the weather patterns and speed analysis to implement the variance based approach to improve the efficiency.

References

[1] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes (2007), „In transit to constant time shortest-path queries in road networks‟, in ALENEX.

[2] R. Bauer, D. Delling, P. Sanders, D. Schieferdecker, D. Schultes, and D.Wagner and D. Wagner, (2010),‟Combining hierarchical and goal-directed speed-up techniques for dijkstra‟s algorithm,‟ ACM Journal of Experimental Algorithmics, vol. 15.

[3] R. J. Gutman (2004), „Reach-based routing: A new approach to shortest path algorithms optimized for road networks‟, in ALENEX/ANALC, pp. 100–111.

[4] B. Jiang (1992), „I/o-efficiency of shortest path algorithms: An analysis‟, in ICDE, pp. 12–19.

[5] Y. Jing, C. Chen, W. Sun, B. Zheng, L. Liu, and C. Tu (2011), „Energy efficient shortest path query processing on air‟, in GIS, pp. 393–396.

[6] N.Jing,Y.W.Huang,andE.A.Rundensteiner(1998) , „Hierarchical encoded path views for path query processing: An optimal model and its performance evaluation‟, IEEE TKDE, vol. 10, no. 3, pp. 409–432.

[7] W.-S. Ku, R. Zimmermann, and H. Wang (2008), „Location-based spatial query processing in wireless broadcast environments‟, IEEE Trans. Mob. Compute., vol. 7, no. 6, pp. 778–791.

[8] G. Kellaris and K. Mouratidis (2010), „Shortest path computation on air indexes‟, PVLDB, vol.3, no. 1, pp. 747–757.

[9] N. Malviya, S. Madden, and A. Bhattacharya (2011), „A continuous query system for dynamic route planning‟, in ICDE, pp. 792–803.

[10] J.Sankaranarayanan,H.Sametandl(2009) Pathoraclesforspatialnetworks‟ PVLDB, vol. 2, no. 1, pp. 1210–1221.

[11] J. Sankaranarayanan and H. Samet, “Query processing using distance oracles for spatial networks,” IEEE Trans. Knowl. Data Eng., vol. 22,no. 8.

[12] H. Samet, J. Sankaranarayanan, and H. Alborzi, (2008) „Scalable network distance browsing in spatial databases‟, in SIGMOD Conference, pp. 43–54.

[13] F. Wei, (2010) „Tedi: efficient shortest path query answering on graphs‟ in SIGMOD Conference, pp. 99–110.