Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 1 (2024)

A Faster Approach for Establishing Trip Similarity

Sudha Chaturvedi
Department of Computer Science and Engineering
Lingayas Vidyapeeth, Faridabad
Sudhachaturvedi18@gmail.com

Dr. Tapsi Nagpal
Department of Computer Science and Engineering
Lingayas Vidyapeeth, Faridabad
dr.tapsi@lingayasvidyapeeth.edu.in

Abstract

Location data traces sourced from devices like GPS, Mobile Signals etc. have important uses in travel pattern applications. Location data traces are arranged as trips and pattern mining algorithms are applied to extract underlying patterns. Many of these algorithms/techniques are dependent on similarity of these trips made by users for example vehicle turn prediction, similarity in route traveling, frequent patterns in traveling the route, user travel route prediction etc. This paper presents an efficient way to establish the similarity between the trips using clustering technique. Proposed technique for trips clustering is efficient in data storage as well as computational time.

1 Introduction

Traffic analysis and user scenario analysis are two most important branches of application which uses trip data [1]. Traffic oriented analysis is used to estimate patterns related to Road networks like travel speed, time dependent use of Roads. User Scenario analysis focuses on mining interesting and useful pattern in user travel behavior.

User travel data is a sequence of GPS traces location $p_0, p_1, p_2 \ldots\ldots p_n$captured from location capturing device. This sequence is a collection of trips. For example a user travels from home to a shopping mall and stays there for some time and return backs to home consists of two trips. First trip is sequence of all the location data for traveling from home to shopping mall $T_0 = p_0, p_1, p_2 \ldots\ldots p_m$ and another trip is for returning from shopping mall back to home $T_1 = p_{m+1}, p_{m+2} \ldots\ldots p_n$. Existing techniques uses these underlying trips for establishing similarity in trips and pattern extraction. These approaches require massive storage in database and are computationally expensive. In this work proposed is to apply a preprocessing step of mapping trip location data to edges of road network using a process called map-matching [3,4] –

$$p_0, p_1, p_2 \ldots\ldots p_n \rightarrow\ e_0, e_1, e_2 \ldots\ldots p_k$$

and then apply trip similarity. This required only storing edges of road network and hence less storage requirement and additionally because of lesser amount of data, computational clustering algorithm runs faster too.

The GPS logs are continuous set of location points. It requires decomposing these set of location points in units of trip. This process is called trip segmentation. Trip segmentation is discussed in section 3.1. Outcome is the set of trips in the form of set of location points. For each trip map matching is applied. It requires Graph of Road network. We discuss Road network graph in section 3.2. Finally algorithm for matching location data to road networks is applied to each trip. Matching process is discussed in section 3.3. Clustering Algorithm for establishing similarity in trips is discussed in section 4.

2. Related Work

Work in [3] considered the trips as a set of location data points and then performed mining to extract frequent patterns in traveling the route. The trips were segmented based on the time gap between the two consecutive trips. Then cells were created around the points. The movements of the objects are represented as transition from on cell to another. All the cells were assumed to be continuous. Whenever any gap between the two cells were reported, interpolation is done and new cells were introduced in the trip. A snapshot of plotting the cells on the map is as shown in the fig 1:

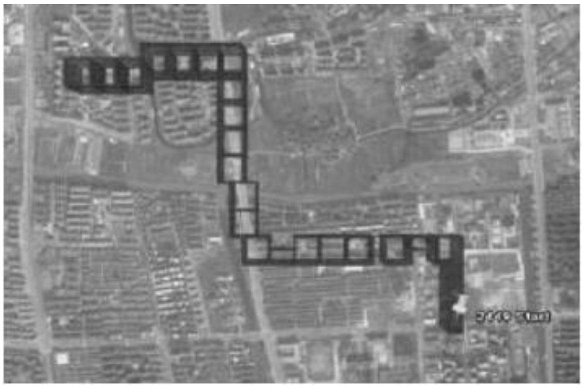


Fig 1: User travel data in cellular zones

This research did not take the structure of the underlying network. This research focused on tracking the route followed by the by the person and not by the vehicle, and hence it can be assumed, the person traveled on the roads. This approach cannot be directly applied for the vehicles which are supposed to follow the roads. Actual road segments travelled cannot be determined.

Authors in [2] also considered the trips as a set of location points and then clustered the similar trips. The similarity between two trips is established as follows:

1. For every location coordinate latitude-longitude pair p_n (x_n, y_n) in trip T_A find the closest trip segment in trip T_B. A trip segment is a line feature between two adjacent temporal location coordinates.

2. Add location segment distances to compute the total distance between trip T_A and trip T_B denoted by $T_{Distance}(T_A, T_B)$.

3. Calculate similarity score $Score(T_A, T_B)$ by dividing $T_{Distance}(T_A, T_B)$ by the number of data points in T_A. This score $Score(T_A, T_B)$ is asymmetric which represents the similarity from T_A to T_B but and may not be same as $Score(T_B, T_A)$ which represents the similarity from T_B to T_A.

4. Repeat the above steps for calculating $Score(T_B, T_A)$ but this time compare trip T_B to trip T_A.

5. Final score is calculated as

$$(Score(T_A, T_B) + Score(T_B, T_A))/2$$

A lower score indicates more similarity.

6. Finally, store the results in a similarity matrix.
The similarity matrix stores the scores as calculated above. Then trips are clustered which are more similar. In the similarity matrix the two trips, with minimum value are merged to form one trip. In each iteration two trips are merged in the same way until some threshold is reached. Thus it is a clustering approach on trip (as set of location points). Means when the similarity score is more than this threshold, further clustering is stopped. The clusters with large number of trips merged, represents the frequently traveled trip.

Any underlying road network was not considered. This approach has following problem:

1) Even when there is no common edges in the two trips but since similarity is computed based on location data only and can end up clustering the two disjoint trips as one trip. This problem can be overcome by proposed process of clustering which is based on performing clustering on edges of road network.

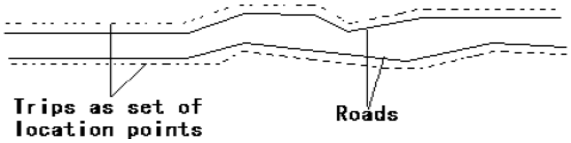


Fig 2: Disjoint trips clustered as one route

2) Whenever it is required to plot the most traveled route, in that case it will be required to map the clustered trips on the map. But this approach looses the information as it is just an approximation of the trips and it will not be possible at all as shown in Fig 2.

Proposed a trip similarity process proposed in [18] is based on text mining of the nodes of the road network names. Each nodes on the road network in named and two trips are considered same when the names of the places in user trips matches.

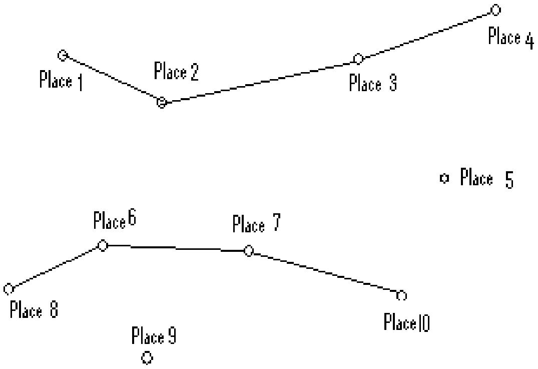


Fig 3: Nodes on the map and data of the two trips

In figure 3, the nodes are shown and also the location points captured are also shown. These Trips will be converted to the textual form and will be output as:

Place1-Place2-Place3-Place4
Place8-Place6-Place7-Place10

Once the trips are in textual from, they applied BM25 Mining algorithm to find similarity between the trips.

But this algorithm too has shortcomings as well. In this approach is based on geometrical comparison only. It is not mentioned as if for any GPS points are equidistant from two nodes, then how to resolve it. The inaccuracies in the GPS data are not taken care off. Although the inaccuracies are inherent due to hardware limitations of the GPS device. By using only geometrical comparisons and in absence of use of topological relations, it may not be possible to map the GPS points to unique nodes on the map in the areas where density of the nodes may be higher. Consider the example below, where the nodes on the map are shown by solid circles and the roads by polylines and the captured GPS points are represented by hollow circles as in Fig 4.

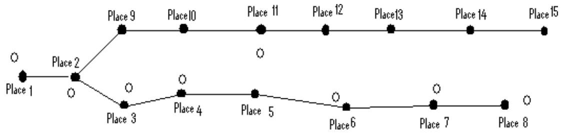


Fig 4: Illustration of shortcomings in Place name based Trip Similarity

It is quite visible that the trip made by the user is

Place1- Place2- Place3- Place4- Place5- Place6- Place7- Place8

But by using the geometrical comparisons and in absence of use of topological relations can create the trip as

Place1- Place2- Place3- Place4- Place11- Place6- Place7- Place8

The mapping of GPS Points also depends upon the neighboring location points as well [4].
If the mapping of the points is incorrect then the errors will propagate to result as well.

3 Conversion of Trips from set of location points to set of edges of Road network

3.1 Trips

GPS devices log the positional information in log files. Example GPS traces stored in database is as shown in Figure 5.

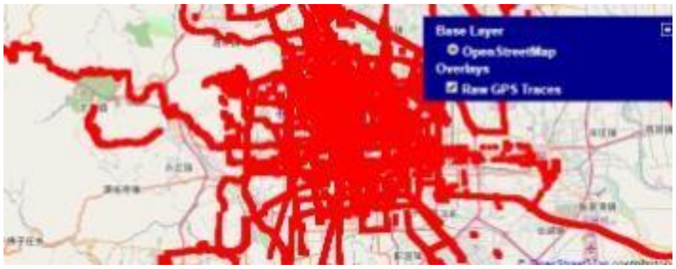


Fig 5. GPS traces stored in database

Decomposing these data into smaller units of trips is known as Trip segmentation. GPS devices log the continuous positional information in log files. Decomposing these data into smaller units of trips is known as Trip segmentation [5, 6]. For example a user travel log contains continuous location ordered sequence $p_1, p_2 \dots . . p_1, p_{i+1} \dots \dots . p_n$ representing user has traveled from $p_1, p_2 \dots . . p_i$ and halts there for some time and then travels $p_i, p_{i+1} \dots \dots . p_n$ then this represents two trips. One realistic scenario representing this is user travels to office in morning and end of day goes to a restaurant is two trips. Further user goes back home after dining then that will be considered as another trip [7]. An example of trip calculated after trip segmentation is as shown in Figure 6.

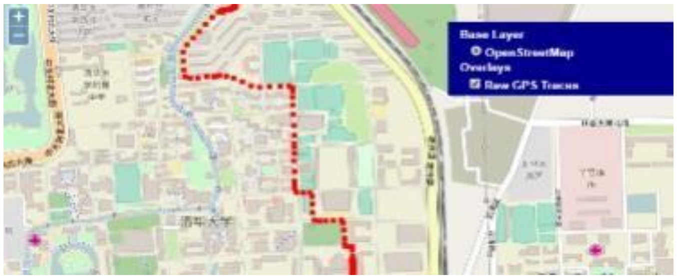


Fig 6. Trip calculated after trip segmentation

3.2 Road Network Graph

Digitized road network is digital representation of real network of any given area. Set of vertices in network are road intersections or important locations in the area. Road segments in the real road network represent the edges of the roads. Road network edges are assigned weights as the length of road segment [6, 8]. Digitized road network is stored in relational database which is spatial enabled. In this work storage is done in Postgres database with PostGis layer enabled to handle geometrical data type. Vertices are represented in database by a point object and road edges are represented by Multiline String [9]. An example of road network data obtained from Open Street Maps (OSM) and stored in database is as shown in Fig. 7.

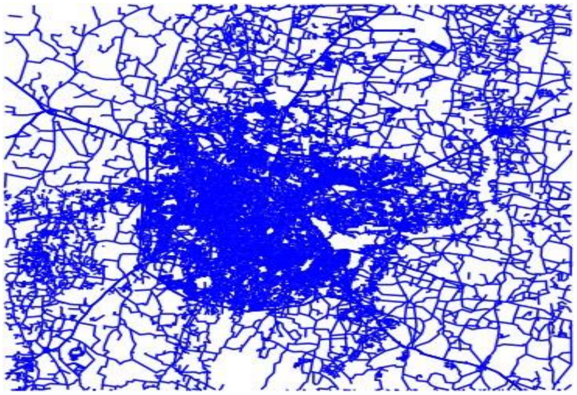


Fig 7. Road network data stored in database

3.3. Mapping of Location Data to Road Network

Research on Map Matching started in 90's and today, Algorithms with high accuracy and efficiency are available. Map Matching deals mapping location traces information to digitized road network [10].

Map matching is a well-established in literature which deals with locating user's location on the road network and has many use cases in different applications which depends on moving object data like route recommendation, traffic prediction, route prediction etc. [11]. When user travels on a road network the GPS location captured may have some inaccuracies which may be because of various factors like error in reading obtained from satellite, hardware limitation of receiving device and in many cases error in digitization of road network. This makes difficult to locate exact position of user on road network and many a times introduces ambiguity in determination process. Map matching process make use of GPS location data and digitized road network and maps user's exact position on road network [12, 13]. In this work, user trips represented by sequence of GPS data points are converted to sequence of road network edges using Map Matching. Fig. 8 shows GPS points in Fig 7 mapped to network and Fig. 9 edges of road network for mapped locations.

Map matching algorithms can be an offline or online process. Offline matching process finds the overall route of the vehicle after the completion of trip by user whereas the online matching process determines user exact location of user on road network in real time. In case of online matching the look-ahead are not available and the algorithm is bound to use only current and previously obtained positional data. While in earlier case look ahead is available [14]. We also adopt offline matching strategy.

Approaches for matching of location data found in literature can be categorized into three groups geometric, Topological and advanced [8]. A Geometric map matching algorithm only considers the digital road network geometry to map a given location coordinate (latitude-longitude) to digitized road network. These algorithms are unaware of topology of road edge segments and only are based on shape of road segment geometry [9, 12]. Map matching algorithms which utilizes geometry of the road segments as well as connectivity relationships of the links is known as topological matching algorithm [10]. Advanced Map matching algorithms uses more complex logic in addition to geometrical and topological information. Some algorithms uses look ahead information, some uses fuzzy logic and some algorithms are developed for handling various kind of uncertainties.

A raw trip composed of GPS location traces is as shown in Figure 8 and resultant data when trip is mapped to road network is as shown in Figure 9.

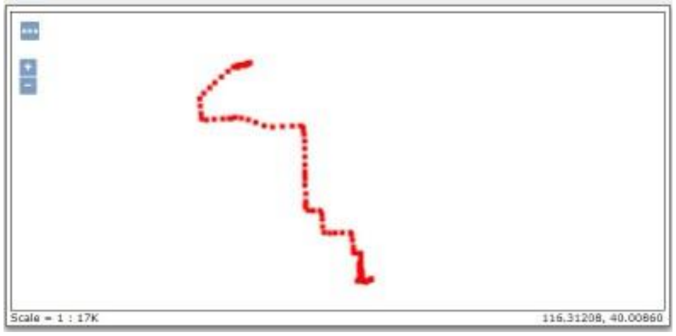


Fig 8. Trip as sequence of GPS coordinates

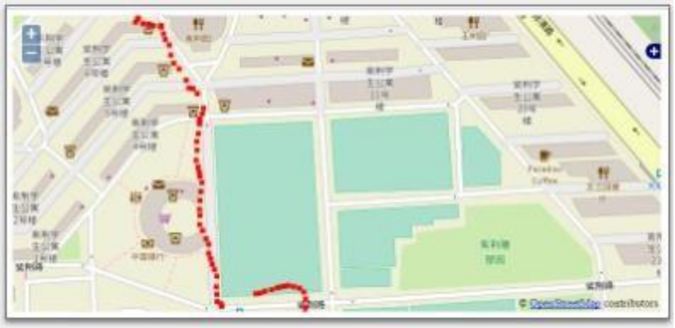


Fig 9. Trip converted to sequence of network edges using map matching

4. Trip Similarity: A Clustering based approach

As described in previous sections, GPS points are segmented as trips (as set of location points) and by use proper map matching algorithm trips are converted to set of edges (of road network). The schema of the table storing the trips is as follows:

Trips_as_edges(id,tripnum,sequence,edge_id,the_geom)

Each Trip is now a set of edges, and hence trip can be represented as string (composed of edge id‘s of the constituent edges). In order to do the clustering of the trips, it requires a metric to calculate the distance between the trips. Unlike the approach of [2] whose distance metric (Hausdroff Distance) clusters the trips which are most spatially closer, we use Leveinstein’s distance to calculate the similarity between the trips. Leveinstein’s distance is used to calculate the similarity between two strings. Unlike approach of [18] where the trips are expressed as strings but strings are composed of names of places and not the path followed between those places. If between two places if more than one path is available, it cannot be distinguished as which path is followed between the two places. In our work the strings represents the segments of the path followed and hence it can distinguish between the different paths followed between the two places even if multiple paths are available.

Distance metric

Levenshtein Distance, is the basic edit distance function whereby the distance is given simply as the minimum edit distance which transforms string1 into string2. Edit Operations are [14]: Copy character from string1 over to string2 (cost 0), Delete a character in string1 (cost 1), Insert a character in string2 (cost 1), Substitute one character for another (cost 1).

For example, the Levenshtein distance is calculated below for the term "BOMBAY" and "MUMBAI", the final distance is given by the bottom right cell, i.e. 3. This score indicates that only 5 edit cost operations are required to match the strings. Strings “BOMBAY” and “MUMBAI” are three distance apart. Using this distance metric, a square matrix representing all pair distances between trips is calculated. This matrix is used to calculate inter-cluster distances while clustering the similar trips.

Clustering of trips

Hierarchical methods is used in this work to perform clustering of trips composed if sequence of road network edges. Clustering starts either start with one cluster and then split into smaller clusters (top down approach) or start with each object in an individual cluster and then try to merge similar clusters into larger and larger clusters (bottom up approach). In this approach, in contrast to partitioning, tentative clusters may be split based on some criteria [15, 17].

The trips clustering naturally have hierarchical structure, so the agglometric clustering technique is used. The agglometric clustering method tries to discover the hierarchy in given datasets. The basic idea of the agglometric method is to start with n clusters for n trips that each trip is a cluster in itself [16]. Using the measure of distance (leveinstein distance), at each step of the method, the method merges two nearest clusters, thus reducing the number of clusters and building successively larger clusters that include increasingly dissimilar objects. In its standard form, the clustering continues until all the data are put into one cluster or is stopped if required number of clusters is obtained. In trip clustering where two trips can have some similarity only if there is some edges are common and also it is not known in advance as how many clusters will be formed. So the clustering is stopped when no two clusters have any trips which have overlapping edges. The method is as described below:

- Allocate each Trip to a cluster of its own. Thus start with n clusters of n trips.
- Create a distance matrix by computing distances between all pairs of clusters using farthest link metric. Sort these distances in ascending order.
- Find two clusters that have the smallest distance between them.
- Remove the pair of objects and merge them.
- If there is no two clusters with trips having some overlapping edges then stop.
- Compute all the distances from the new cluster and update the distance matrix after the merger and go to Step 3.

For example let the trips be,

Trip1: 17619, 17617, 17615, 17613, 17611, 176090
Trip2: 17619, 17617, 17615, 17613, 17611, 32195, 32197
Trip3: 17617, 31961, 32167, 32165, 32163, 32063, 32061
Trip4: 32255, 32253, 17617, 17615, 31933, 31935, 31937
Trip5: 32255, 32253, 17617, 17615, 31933, 31935, 32265

The initial distance matrix thus formed is as follows

	[1]	[2]	[3]	[4]	[5]
[1]	0.0	2.0	7.0	5.0	5.0
[2]	2.0	0.0	7.0	6.0	6.0
[3]	7.0	7.0	0.0	7.0	7.0
[4]	5.0	6.0	7.0	0.0	1.0
[5]	5.0	6.0	7.0	1.0	0.0

Each trip is a distinct cluster. So there are 5 clusters:
[1] [2] [3] [4] [5]

Clusters at the minimum distance are [1] and [2] so they are merged to form a new cluster [1 2], and the new distance matrix is:

	[1,2]	[3]	[4]	[5]
[1,2]	0.0	7.0	6.0	6.0
[3]	7.0	0.0	7.0	7.0
[4]	6.0	7.0	0.0	1.0
[5]	6.0	7.0	1.0	0.0

Clusters at the minimum distance are [4] and [5] so they are merged to form a new cluster [4 5], and the new distance matrix is:

	[1,2]	[3]	[4,5]
[1,2]	0.0	7.0	6.0
[3]	7.0	0.0	7.0
[4,5]	6.0	7.0	0.0

Clusters at the minimum distance are [1 2] and [4 5] so they are merged to form a new cluster
[[1 2][4 5]], and the new distance matrix is:

	[[1,2][4,5]]	[3]
[[1,2][3,4]]	0.0	7.0
[3]	7.0	0.0

Clusters at the minimum distance are [[1 2][4 5]] and [3] so they are merged to form a new cluster

[[[1,2][4,5]][3]]

And here the clustering stops.

This Clustering can be represented graphically as:

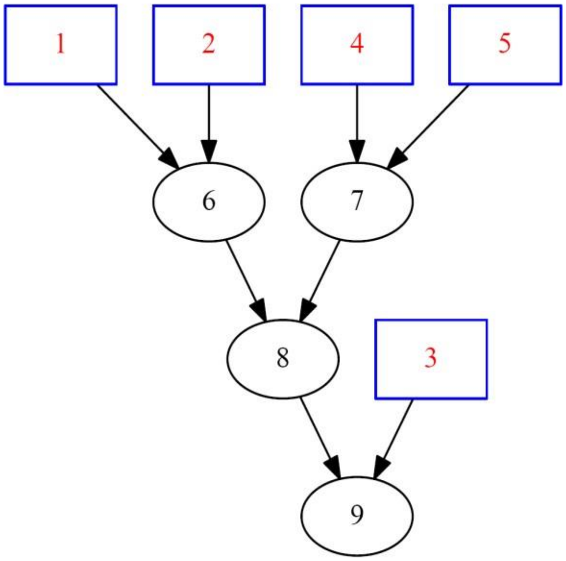


Fig 10: Graphical representation of the

Proposed Trip clustering Algorithm
Text representation of the Agglometric tree is as follows:

Node	**Cluster**
1	[1]
2	[2]
3	[3]
4	[4]
5	[5]
6	[1,2]
7	[4,5]
8	[[1,2][4,5]]
9	[[[1,2][4,5]][3]]

5. Implementation

We took Road Related data from Open Street Maps (OSM). Road network data is sourced from Open Street Map (OSM). OSM is open platform from where digital data like road network, land usage, water bodies, national and international boundaries can be downloaded. In this work road network data from India and China is used. Data is stored in geo-Spatial enabled data relation base. For this purpose Postgres database with PostGis data layer enabled for handling geometrical data types. Road network data is preprocessed in Sql format and imported in data base.

Osm2pgsql is a utility with Postgres that can be used to load the content of the OSM file to Postgres database. For each type of the feature like Road lines, polygons etc., there is a separate file. It is required to load each of these files separately into the database. The OSM file containing all the data of the whole world is of 15 GB in compressed form. The size of the file corresponding to different levels may be of different size.

Osm2po is a utility which takes OSM files as input and generates a routable map. The resultant of the tool is a SQL file which can be directly loaded to Spatial enabled database in our case PostGis [5, 6]. In our algorithm we used this stored graph to find the topologically connected edges to the current edge in consideration.

In this experiment we used GPS data released by Microsoft collected during the Geolife project. This data is collected over a span of 2 years and mainly from Beijing area of china [19, 20, 21].

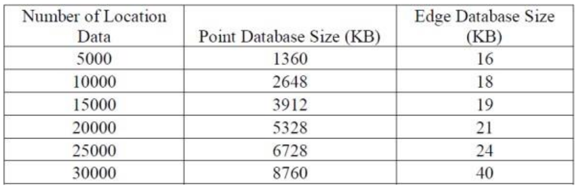
This data is a continuous log of GPS latitude longitude position.
The data collected from location acquisition devices consists of positional information and time stamp. The data collected for this project has the following format:

< vehicle, Latitude, Longitude, time, Date>

We decomposed it into trips for analysis. Then Map Matching is applied to these Trips. Once the trips are converted as set of network edges, Clustering algorithm discussed above. Results are as in following section.

Results

The project was evaluated on both the measures- Space and time, and it was observed that it performs well on both the measures as compares to other implementations. Running time of clustering is as captured in Table 2 & 3 and visual representation as in Fig 11 & 12.



Number of Location Data	Point Database Size (KB)	Edge Database Size (KB)
5000	1360	16
10000	2648	18
15000	3912	19
20000	5328	21
25000	6728	24
30000	8760	40

Table 2 Growth of database size

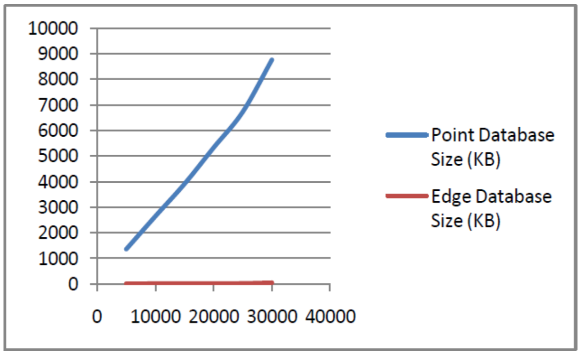
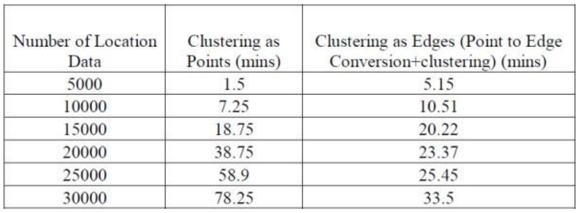


Fig 11: Graph of Table size



Number of Location Data	Clustering as Points (mins)	Clustering as Edges (Point to Edge Conversion+clustering) (mins)
5000	1.5	5.15
10000	7.25	10.51
15000	18.75	20.22
20000	38.75	23.37
25000	58.9	25.45
30000	78.25	33.5

Table 3 Running time of clustering algorithms on cluster of multiple nodes

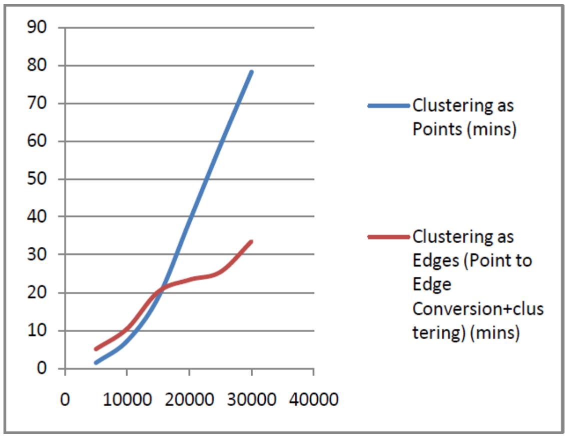


Fig 12: Graph of running time of clustering algorithms

Conclusions

The location data collected over a considerable span of time has huge number of travel patterns hidden in it. Those patterns cannot be extracted through querying the databases. If this data is subjected to appropriate Data Mining techniques, these patterns can be extracted. Before mining can be done, the data needs to be preprocessed properly. This includes the organizing the location point data in the forms of trips. Point data of trips can be subjected to map-matching process to convert trip as set of edges of graph of road network. Clustering of trips (as set of edges) can then be done by agglometric clustering technique.

GPS traces from Microsoft Geolife Project [21] are used in this project to form the trips as set of location points. Road network database is constructed from the arcs and nodes downloaded from Open Street Map. Location data of each trip is associated to edges of road network to convert trips as set of location point into trips as set of edges. This work compared the performance of trip clustering algorithm when trips are expressed as points and when trips are expressed as edges. It was observed that, the approach of trip cluster by first computing trips as edges and then doing clustering is efficient in both the terms, space as well time.

References:

[1] Yue,Y, Zhuang,Y,Li,Q, Mao,Q, Mining time-dependent attractive areas and movement patterns from taxi trajectory data, 2009 17th International Conference on Geoinformatics, Fairfax, VA, ISBN: 978-1-4244-4562-2,Aug 2009.

[2] Froehlich,j., Krumm,j.,Route Prediction from Trip Observations,SAE 2008-01-0201.

[3] Ye, Q., Chen, L. & Chen, Gc. Personal continuous route pattern mining. J. Zhejiang Univ. Sci. A 10, 221–231 (2009). https://doi.org/10.1631/jzus.A0820193

[4] White, C.E., Bernstein, D., Kornhauser, A.L. ,2000, Some map matching algorithms for personal navigation assistants. Transportation Research Part C 8, 91-108.

[5] Tiwari,V.S., Arya,A., 2018, Distributed Context Tree Weighting (CTW) for Route Prediction, Open Geospatial Data, Software and Standards, Springer Journal, 2018.

[6] Tiwari,V.S., Arya,A., 2017, Horizontally scalable probabilistic generalized suffix tree (PGST) based route prediction using map data and GPS traces, Journal of Big Data, Springer Journal, 2017.

[7] Nawaz, A.; Zhiqiu, H.; Senzhang, W.; Hussain, Y.; Naseer, A.; Izhar, M.; Khan, Z., 2020, Mode Inference using enhanced Segmentation and Pre-processing on raw Global Positioning System data, Meas. Control 2020

[8] Ekim, B., Sertel, E., & Kabadayı, M. E., 2021, Automatic road extraction from historical maps using deep learning techniques: A regional case study of turkey in a German World War II Map. ISPRS International Journal of Geo-Information, 10(8), 492

[9] Saeedimoghaddam, M., Stepinski, T. F., 2020, Automatic extraction of road intersection points from USGS historical map series using deep convolutional neural networks. International Journal of Geographical Information Science, 34(5), 947-968.

[10] Quddus, M. A., Noland, R. B., Ochieng, W. Y., 2006, A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport, Journal of Intelligent Transportation Systems, 10: 3, 103 — 115

[11] Yin, H., Wolfson,O., 2004, A Weight-based Map Matching Method in Moving Objects Databases, SSDBM '04 Proceedings of the 16th International Conference on Scientific and Statistical Database Management IEEE Computer Society Washington, DC, USA, ISBN:0-7695-2146-0

[12] Lou,Y., Xie,X., Zhang,C., Wang,W., Zheng,Y., Huang,Y., 2009,Map-Matching for Low-Sampling-Rate GPS Trajectories , ACM GIS ISBN 978-1-60558-649-6/09/11

[13] Nawaz, A., Huang, Z., Senzhang, W., Yasir, H., Izhar, K., Zaheer, K., 2020, Convolutional LSTM based transportation mode learning from raw GPS trajectories. IET Intelligent Transport Systems. 14. 570-577. 10.1049/iet-its.2019.0017.

[14] Srivastava, K., Pandey P.C., Sharma, J.K., 2020, An approach for route optimization in applications of precision agriculture using UAVs. Drones 4(3), 58 (2020)

[15] Gao, Caroline X; Dwyer, Dominic; Zhu, Ye; Smith, Catherine L; Du, Lan; Filia, Kate M; Bayer, Johanna; Menssink, Jana M; Wang, Teresa; Bergmeir, Christoph; Wood, Stephen; Cotton, Sue M, 2023, An overview of clustering methods with guidelines for application in mental health research, Psychiatry Research, ISSN: 0165-1781, Vol: 327, Page: 115265

[16] Chen Zhang, 2022, Research on Literature Clustering Algorithm forMassive Scientific and Technical Literature Query Service, Computational Intelligence and Neuroscience Volume 2022, Article ID 3392489, https://doi.org/10.1155/2022/3392489

[17] Barbara Probierza, Jan Kozaka, Anita Hrabia, 2022, Clustering of scientific articles using natural language processing, 26th International Conference on Knowledge-Based and Intelligent Information & Engineering, Systems (KES-2022). Procedia Computer Science 207 (2022) 3443–3452

[18] Doherty,A.R, Gurrin,C., Jones, G.J.F., Smeaton,A.F., Retrievl of similar Travel Routes Using GPS Tracklog Place Names, SIGIR GIR'06, August 10,2006, Seattle, USA

[19] Zheng, Y., Zhang, L., Xie X., Ma, W. Y., 2009, Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of International conference on World Wild Web (WWW 2009), Madrid Spain. ACM Press: 791-800

[20] Zheng., Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y., 2008, Understanding Mobility Based on GPS Data. In Proceedings of ACM conference on Ubiquitous Computing (UbiComp 2008), Seoul, Korea. ACM Press: 312-321.

[21] Zheng, Y., Xie, X., Ma, W.Y., 2010, GeoLife: A Collaborative Social Networking Service among User, location and trajectory. Invited paper, in IEEE Data Engineering Bulletin. 33, 2, 2010, pp. 32-40.