ISBN 978-981-14-4787-7
Proceedings of 2020 the 10th International Workshop on Computer Science and Engineering (WCSE 2020)
Yangon (Rangoon), Myanmar (Burma), February 26- February 28, 2020, pp. 151-155
doi: 10.18178/wcse.2020.02.026

GPS Trajectory Cleaning For Driving Behaviour Detection System

Tin Lai Lai Mon[+]

University of Computer Studies, Yangon, Myanmar

Abstract. 2017 records of WHO show that road traffic accidents deaths in Myanmar reached 10,527 or 2.67% of total deaths. It also shows that most of the road traffic accidents are due to drivers. Therefore, it is essential to develop a system which is able to detect the drivers' driving behavior to reduce the traffic accidents. Nevertheless, in order to detect the drivers' driving behavior, their accurate GPS trajectory is needed and there are still lots of challenges to get accurate GPS trajectory. GPS trajectory cleaning is one of the most important steps for preprocessing GPS data. GPS data might be wrong because sometimes signals transmitted by satellites cannot be recorded accurately by GPS receivers because of the interference, weak signal, and malfunction of sensor. Therefore, the recorded GPS point can be far from the actual location of the receiver. This inaccurate location can make strong affect to some decision making processes based on the GPS data. In order to eliminate wrong GPS data points from the trajectory, some trajectory cleaning processes: detecting stop points, interpolating missing segments and removing inaccurate points are proposed in this paper. Moreover, the results show that preprocessing trajectory cleaning approach helps to improve the quality of trajectory clustering.

Keywords: Trajectory cleaning, Detecting stop points, Interpolating missing segments, Trajectory clustering.

1. Introduction

Clustering is a processing of grouping similar objects. GPS data can be points or trajectory data. By clustering trajectory data, one can find clusters of objects that follow the same path or detect groups that moved together for a period of time. One can also detect a driver's driving behavior from his trajectory. If it is able to detect a driver's behavior, it is also possible to prevent from road traffic accidents due to drivers by alerting the driver to change his driving behavior at once. But, in order to correctly detect his driving behavior, the accuracy of the trajectory data is crucial. Although there are many methods of trajectory clustering, there is a lack of data preprocessing which is very important to get the result with high accuracy. Critical roles of trajectory data mining in modern intelligent systems are surveillance security, abnormal behavior detection, crowded behavior analysis and traffic control. Clustering algorithms can be categorized into three. They are unsupervised, supervised and semi-supervised algorithms.

In this paper, there will be three steps for cleaning trajectories: stop detection, missing segments and interpolation and inaccuracy removal. Trajectory preprocessing is evaluated by clustering both raw and the cleaned dataset and comparing results. And the results show that the proposed preprocessing gives the better quality of clustering.

2. Proposed System

2.1. Overall System Design

Our proposed overall design of driving behavior detection system is shown in Figure 1. There are 4 stages in our system. The first stage is to collect GPS data, the second stage is to do trajectory cleaning, the

[+] Corresponding author. Tel.: +959-95300903
E-mail address: tinlailaimon@ucsy.edu.mm

third stage is to enhance data by adding weather conditions, road speed limit and important places and the last stage is to measure the aggressiveness of a driver [1]. In this paper, the second stage, trajectory cleaning, will be emphasized.

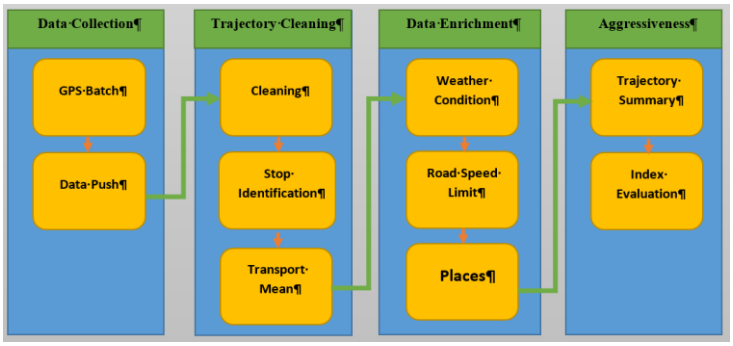


Fig. 1: Overall Design of Driving Behaviour Detection System

2.2. Clustering Methods

Clustering methods can be divided into three main groups. They are model-based clustering; distance-based clustering and visual-aided clustering [2]. In model-based clustering data is considered as coming from a distribution that is mixture of two or more clusters. Each cluster corresponds to a different distribution, and generally, the distributions are assumed to be Gaussians. The parameters of each distribution are estimated by maximizing the likelihood of the expression data. The k-mean approach is a special case of model-based clustering, where all the distributions are assumed to be Gaussians with equal variance. Distance-based methods use distance functions to show similarity between objects. This allows breaking the whole trajectory clustering process into two steps: (1) calculation of distances between trajectories according to the defined distance function and (2) actual clustering using a known clustering algorithm. Finally, visual-aided methods depend on the decision of experts. Experts will change the clustering settings according to their knowledge and experience until the system achieves the desired result [3].

2.2.1. DBSCAN Method

In this work, one of the well-known clustering algorithms, a density-based clustering algorithm called DBSCAN is used to classify every GPS point as CORE, BORDER or NOISE. DBSCAN classification is based on the input parameters: minimum points to cluster (*minPts*) and the radius (*Eps*). At first, *minPts* is necessary to define and every point which has at least *minPts* in radius (*Eps*) can be classified as CORE. A BORDER is a point which does not satisfy the CORE criteria but it is connected to a CORE. Every point which is neither a CORE nor a BORDER is classified as NOISE. An example of a DBSCAN clustering is shown in Figure 2. In this figure, *minPts* =4. Point A and the other red points are core points because the area surrounding these points in the radius (*Eps*) contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are also included in the cluster as they are BORDER points. Point N is a NOISE point that is neither a CORE point nor BORDER. This algorithm will be modified in the stop detection step and during actual trajectory clustering.

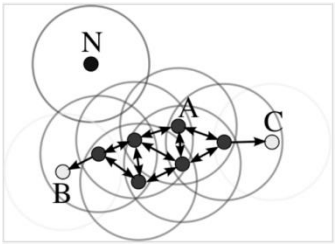


Fig. 2: DBSCAN clustering

2.2.2. Dynamic Time Warping (DTW)

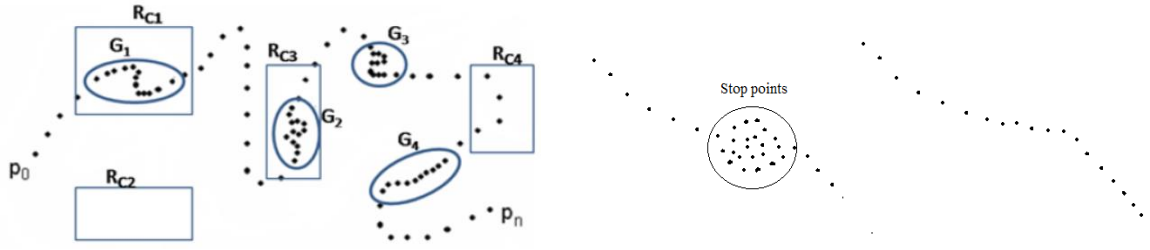
One of the algorithms for measuring similarity between two temporal sequences used for GPS trajectory data is Dynamic Time Warping (DTW). DTW was originally developed for speech recognition. DTW is used to calculate the similarity between objects that have different speeds. Suppose that there are two

sequences of time series. By using DTW, all points of these sequences are warped to each other to minimize the resulting distance. The main advantage of DTW in comparison with Euclidean distance function is that it includes stretching and compression of sequences. In this paper, DTW is used as a distance function that data preprocessing can improve clustering. However, there are also other distance functions to be used.

2.3. Stop Detection

There are many stop detection methods. One of those methods called Intersection-Based Stops and Moves of Trajectories (IB-SMoT) is considered based on the intersection of a trajectory with the user-specified relevant feature types (interesting area or building) for a minimal time duration. Suppose that a trajectory intersects the geometry of an interesting area or building and if the duration of this intersection is greater than or equal to the predefined amount of time, it is considered as a stop point.Another method called Clustering-Based Stops and Moves of Trajectories (CB-SMoT) finds stops nearby places on the trajectory where object spent a relatively large time without leaving those locations. In this work CB-SMoT is applied to find stops and remove them. The purpose of removing these stops is that many distance functions are quite sensitive to them. For example, DTW processes each point in a sequence, the distance between trajectory A (without stops) and similar trajectory B (with stops), would be larger compared to distance between trajectory A and trajectory C (without stops from trajectory B). This may lead to decrease accuracy of trajectory clustering.

In this research work, to detect stop detection three parameters *Eps* which is used in DBSCAN method, *minTime* (time an object spent at particular locations) and area (approximate proportion of points that can form stops) will be selected automatically. In order to detect stop point, it is necessary to find core point condition first. A core point can be defined based on *Eps* and *minTime*, if $minTime \leq T_{last} - T_{first}$, where T_{last} is the latest timestamp and T_{first} is the earliest timestamp in the *Eps*-neighborhood of the core point. The cluster is expanded with all the density-reachable points from the *Eps*-neighborhood. According to the observation, it is found that *Eps* parameter can be easily estimated by taking the mean of all the distances between consecutive points of a trajectory and this mean is sufficient to detect all the major stops on a trajectory. Founded stops are shown in Figure 3(a) where p0,p1,…,pn are GPS points, G1,G2,G3 and G4 are the clusters and RC1, RC2, RC3 and RC4 are candidate stops [4]. After finding all the stops on the trajectory, the stops are moved and filled in the created gap with points calculated based on Missing Segment Interpolation. The comparison of GPS points before and after removing stop points is shown in the following Figure 3(b).



(a) (b)

Fig. 3: (a). Stop detection, (b). Before and after removing stop points

2.4. Missing Segment Interpolation

In trajectory data, there may be some gaps because of the loss of GPS signal. Then, it is necessary to estimate missing segments (missing part of a trajectory according to a given GPS sampling rate and object's movement direction). In this work, missing segments are emulated using a simple interpolation technique.

It is obvious that in order to fill the missing gap it just needs to connect the closest two points before and after the perceived gap. But such kind of consideration leads to the ignorance of the GPS data sampling rate. Moreover, there will be inaccurate results in finding DTW distance.

Suppose P_t and P_{t+1} are consecutive points on a trajectory with timestamps t_{Pt} and t_{Pt+1} and let φ and ψ be the interpolation and trajectory breaking thresholds, respectively. If the time difference between t_{P1} and t_{P+1} is larger than the interpolation threshold φ, this segment is said to be "complete". If this difference is also greater than the second threshold ψ, this segment will not be interpolated. Instead, the segment will be

broken down into two separate trajectories. Moreover, given trajectory will be separated into two sub trajectories if there is no GPS signal for ψ amount of time. This trajectory partitioning is performed even before stop detection. Suppose that

P – the list of points on a trajectory
P_a, P_b – the endpoints of the missing segment and
N – the number of sub segments.

Then, $N = \lceil \frac{2kDist(Pa,Pb)}{\sum_{j=a-k}^{a-1} Dist(P_j,P_{j+1}) + \sum_{j=b}^{b+k-1} Dist(P_j,P_{j+1})} \rceil$

where, $Dist(P_a,P_b)$ is the Euclidean distance between Pa and P_b.

Then, the distance between two consecutive generated points P_i and P_{i+1} (where $a < I < b$) to fill in a missing segment is defined as:

$$Dist(P_i, P_{i+1}) = \frac{Dist(P_a, P_b)}{N}$$

After that, N-1 points are created starting from Pa towards P_b according to the calculated distance $Dist(P_i, P_{i+1})$. After generating all required points, a timestamp for each point is added based on the number of generated segments and on the time difference between two endpoints of the missing segment. In this way, missing segments will be filled using the interpolation algorithm mentioned above. Before and after interpolation of missing segments is shown in Figure 4.

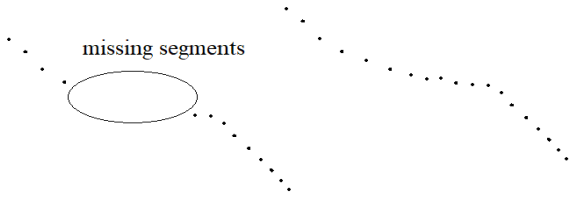


Fig. 4: Before and after doing missing segment interpolations

2.5. Removing inaccurate GPS Points

Some erroneous points exist in many datasets and they are also needed to remove. Erroneous points may be points which have the same coordinates but separated in time. Such kind of erroneous points cannot be found by stop detection if their timestamps is satisfied with *minTime* threshold value. So, the points that have the same coordinates are needed to remove. Some trajectories with null speed and randomly changing their locations with time are also needed to be removed. Moreover, after breaking down the raw dataset into separated trajectories, trajectories which give no meaningful knowledge to the final clustering are also needed to be removed. Therefore, trajectories that did not meet the threshold on the minimum number of points per trajectory are eliminated.

3. Experimental Results

This research proposal is implemented based on the raw trajectory data. From the raw data, sub trajectories are extracted. And then, those sub trajectories are preprocessed using stop detection, missing segment interpolation and inaccurate points removing. And then, there are two datasets: raw dataset and cleaned dataset. Similarity matrices of both datasets are computed using DTW distance between trajectories. Then computed similarity matrices are inputted to DBSCAN. Finally, clustering results of both dataset are compared using a quality measure, QMeasure. To calculate QMeasure the Sum of Squared Error (SSE) and the noise penalty which is to penalize incorrectly classified noises are used. QMeasure is used to minimize the sum of squared pairwise distances between elements that belong to one cluster (Total SSE), while penalizing for incorrectly identified noise points:

$$QMeasure = TotalSSE + NoisePenalty$$
$$= \sum_{i=1}^{|C|} \frac{1}{2|C_i| \sum_{x \in C_i} dtwDist(x,y)^2} + \frac{1}{2|F| \sum_{w \in F} \sum_{z \in F} dtwDist(w,z)^2}$$

where, C is a set of clusters C_i, F is a noise trajectories set and *dtwDist(x,y)* is the DTW distance between trajectories x and y. In this case, *QMeasure* with smaller values means more accurate clustering.

Comparison of *QMeasure* for both dataset of two trajectories is shown in the following Figure 5. In Figure 5(a) and 5(b), Eps value is fixed to 2000 and QMeasure is calculated by changing the minPts from 0 to 10 for Trajectory-1 and Trajectory-2. In Figure 5(c) and 5(d), minPts is fixed to 3 and QMeasure is calculated by changing the Eps values for the Trajectory-1 and Trajectory-2.

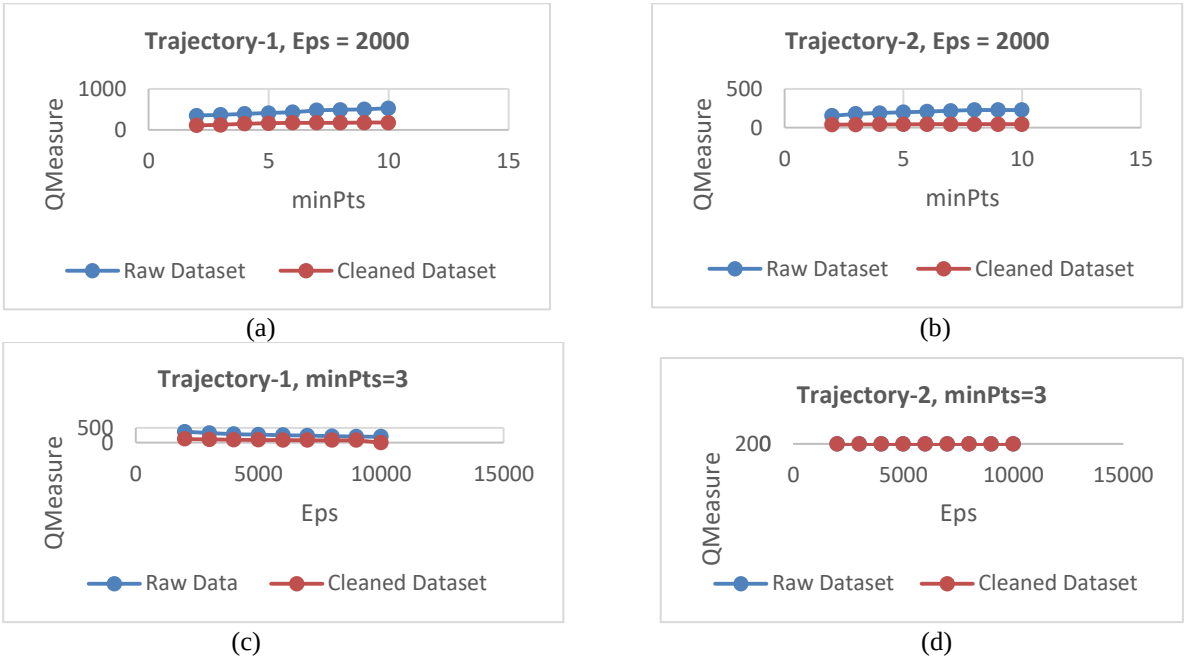


Fig. 5: Comparison of QMeasure between raw dataset and trajectory cleaned dataset

According to the experimental results, it is obviously seen that for any trajectory path with changing *minPts* values and *Eps* values, clustering quality of the proposed cleaned dataset is always better.

4. Conclusion

In this paper, the second stage of the research work, a trajectory cleaning process using trajectory clustering methods: DBSCAN and CB-SMoT, is proposed. Before clustering, GPS dataset is cleaned using stop removal, missing segment interpolation and inaccurate point removal. The clustering quality measure shows that clustering dataset only after cleaning using the proposed method has more accurate clusters.

In our future work, it is expected to develop a trajectory clustering process which works automatically and gives more efficient results. For stop detection, users still need to give *minTime* value. In finding the missing points using segment interpolation, there is still needed to find a smoother interpolation method so that the estimated points are much closer to the real points. Cleaned trajectories will be used in the third stage of the research work, data enrichment, by combining the weather conditions, road speed limit and important places.

5. References

[1] Tin Lai Lai Mon, Thin Lai Lai Thein, "Design and Implementation of Smart Alert System for Reducing Road Traffic Accidents in Myanmar", *AIP Conference Proceedings 2129*, 2019.

[2] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo, "Spatio-temporal clustering", Data Mining and Knowledge Discovery Handbook, second edition, Springer, 2010.

[3] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti, "Interactive visual clustering of large collections of trajectories", *Visual Analytics Science and Technology*, pages 3 –10, 2009.

[4] L. O. Alvares, G. Oliveira, C. A. Heuser, and V. Bogorny, "A framework for trajectory data preprocessing for data mining", *Int. Conf. on Software Engineering and Knowledge Engineering*, pages 698 – 702, 2009.