Stop Point Identification Using Constrained DBSCAN (Density-Based Spatial Clustering of Applications with Noise) Algorithm

Lei GONG[1], Hitomi SATO[2], Tomio MIWA [3] and Takayuki MORIKAWA[4]

[1]Member of JSCE, Doctoral Student, Dept. of Civil Eng., Nagoya University
(Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan)
E-mail:Leigongchi@gmail.com
[2]Member of JSCE, Designated Lecturer, Institute of Innovation for Future Society, Nagoya University
(Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan)
E-mail: sato@trans.civil.nagoya-u.ac.jp
[3]Member of JSCE, Associate Professor, EcoTopia Science Institute, Nagoya University
(Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan)
E-mail: miwa@nagoya-u.jp
[4]Member of JSCE, Professor, Institute of Innovation for Future Society, Nagoya University
(Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan)
E-mail: morikawa@nagoya-u.jp

GPS data provide a substitute method of obtaining Personal Trip (PT) data for traditional survey such as pencil-and-paper interview. Trip segmentation is usually the first step of obtaining PT data from continuous trajectories and stop point identification is essential in segmenting trips. In this paper, we propose a constrained DBSCAN (ConstDBSCAN) algorithm for identifying stop points from a series of GPS track points. ConstDBSCAN algorithm is a spatial density-based clustering methods temporally and directionally considering features of tracking points in continuous trajectories. Compared to other variants of DBSCAN algorithm, ConstDBSCAN advanced in this paper achieves a higher accuracy of 90%.

Key Words : *stop point identification, density-based, DBSCAN, GPS trajectory*

1. INTRODUCTION

Personal Trip (PT) data are essential to transportation demand analysis in a city or region. Compared to traditional methods of obtaining PT data, using GPS points can avoid the problem such as inaccurate trip time and low trip rate. However, trip segmentation is the fundamental step before obtaining trip purpose and transportation mode. Trip is usually segmented by a series of activities and these activities can be identified by difference factors according to various methods[1)].

In this paper, we advance a constrained DBSCAN (ConstDBSCAN) algorithm for identifying stop points from a series of GPS track points. The remainder of this paper is arranged as follows. Section 2 reviews related researches. Then data set used in this paper is interpreted in section 3. It is followed by section 4 where methodology developed in this paper is advanced. Results of applying this methodology to dataset in section 3 are given in section 5. Section 6 shows a comparison of ConstDBSCAN with other variants of DBSCAN. Finally, conclusions are drawn in section 7.

2. RELATED RESEARCH

Trajectories are a series of GPS points definitely with location and time, additionally with speed, acceleration and so on. So far these features of trajectories have been used directly or indirectly in existing methods to identify stops. These methods can be generally categorized into following groups: duration-based method, density-based method, velocity-based methods centroid-based method and hybrid method (hybrid method usually uses two of the features in the former methods). A summary of existing methods is listed in Table 1.

.

Table 1 Summary of methods for identifying stop from GPS data

Paper	Category	Methods/Algorithms	POI info. Involved	Data set	Accuracy
2	centroid-based	k-means clustering algorithm	no	one person, 4 months, in Atlanta, Georgia	---
12	density-based	Density-and-Join-based (DJ-cluster) algorithm, a simplified edition of DBSCAN	no	28 respondents, 3 weeks, interval: 1 min	85%
9	duration-based	Stops and Moves of Trajectories (SMoT) algorithm, which uses geographic information to identify the candidate stops as the input of SMoT algorithm	yes	---	---
7*	duration-based	Clustering-Based SMoT (CB-SMoT) algorithm, a variant of DBSCAN	no	students as respondents in Amsterdam(125000 GPS points, 487 trajectories)	---
3	speed-based	scoring function based on speed	no	mining haulage vehicles (130 thousand samples)	---
10	density-based	Trajectory Ordering Points To Identify the Clustering Structure (T-OPTICS), a variant of T-OPTICS	no	one person, 7 hours by walking, cycling and driving a car, interval: 1-4s	---
8	duration-based	Point Of Interest Activity Mapping Set (PAMS)	yes	simulated trajectories	---
13	hybrid method	speed-and-duration-based approach with dynamic speed threshold related to average speed of current moving object and average speed of moving objects in this position	no	Car dataset (Milan, 17241 objects, 2075213 points, 1week, interval: avg. 40s). Bus dataset (Athens, 2 objects, 66095 GPS points, 108 days, interval: 30s). Truck (Athens, 50 objects, 112203 GPS points, 33 days, interval: 30s). Taxi (Lausanne, 2 objects, 3347036 GPS points, 5 months, interval: 1s)	---
5	duration-based	10min as duration threshold	no	car GPS data (young drivers, 119 cars,15months, 0.1billion GPS records)	---
11	density-based	fast density-based probabilistically algorithm	no	One person in car in Miyako Japan. Data interval: 15 sec, totally 1617 data points	---
6*	duration-based	Trajectory DBSCAN (TrajDBSCAN) algorithm	no	Nokia dataset (6 users, 2324 trajectories, 178667 GPS points, interval: 10s). Milan dataset (4162 private cars, 5749 trajectories, 190779 GPS points, interval: 10~600s)	---
14	hybrid method	duration and distance based criteria (5 min and 100m)	no	one person, 351 days, 81389 GPS positions	---
4	speed-based	use speed and change rate of average speed as vector in Support Vector Machines	no	3 persons, Hakodate	98.8%(cross validation)
15	hybrid method	duration and distance based criteria (5min as duration threshold; 25m as distance threshold) to split trip chains into separate trips	no	8141 persons, 362 cars, 3.2 million GPS points	---

Note: POI stands for Point of Interest.

*since the concept of core points was changed by replacing minimum number of point in a neighborhood by minimum duration in a neighborhood, in this table, these two variant of DBSCAN algorithm are categorized as duration-based methods instead of density-based methods.

A **centroid-based method**, specifically, a variant of k-means clustering algorithm was applied in research[2] by iteratively calculating mean of points (new centered point) within a given radius of temporary centered point (the centroid of points in the radius) until the centered point in the radius of points does not change any more. However, k, the number of clusters has to be known beforehand. It is nearly impossible to know how many stops there are in a series of trajectories.

Speed-based methods were proposed in research[3,4]. Research[3] invented a scoring function involving speeds to reflect the significance of vehicles' current location. The scoring function defined the significance of current location by comparing current speed to two speed thresholds and in a mining environment. Research[4] used speed and change rates of average speed as input features in SVMs to obtain the move and stop point. Actually speed-based methods need to know speed which is not always applicable to all GPS devices or modules. Besides, some limitation occurs in the situations such as objects moving in parking lot or stuck in a traffic jam or bad weather conditions.

Duration-based methods are the most popular and can be found in research[5, 6, 7, 8, 9]. Research[5] extracted stay points by judging the duration between two consecutive records from a user is larger than a threshold of 10 minutes. Research[6, 7] applied a modified DBSCAN algorithm by using minimum stop duration instead of minimum number of points in a neighborhood when defining core points. The difference is that straight distance between two points was used for distance calculation in research[6] while distance along trajectory is used for distance calculation in research[7]. Researches[8, 9] identified stops by judging stop duration and whether the GPS point intersects the geometry of a spatial location. The difference is that research[8] utilized a matching table containing minimum and maximum elapsed time for each possible type of activity to map the trajectory to possible activities whereas research[9] used a given threshold stop duration. One problem of duration-based methods is to decide optimal duration threshold because the result is very sensitive to this threshold.

Density-based methods can be found in research [10, 11, 12]. Research[10] utilized an interactive density-based clustering algorithm, in which the density was defined on the basis of both the spatial and the temporal properties of a trajectory. Research[11] proposed a fast algorithm for probabilistically extracting significant locations from raw GPS data based on data point density. This algorithm eases the difficulty in parameter setting and works well even if there are a variety of noise levels in input data. Research[12] used density-joinability to simplify the mechanism of expanding cluster in DBSCAN. According to the simplified mechanism, any sharing point in any two clusters can be joined together as one cluster. Methods of this type need data in a more frequent interval. Moreover, since density-based methods use the concept of spatial points clustering methods, adjustments are needed when applied in GPS trajectory situation.

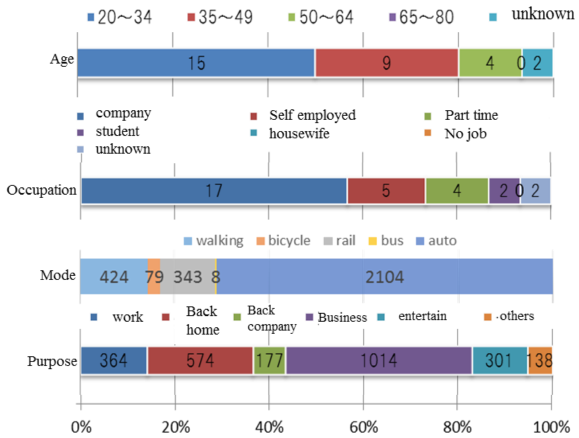
Hybrid method used two of the variables such as speed, duration, density etc. together. Research[13] used speed threshold and minimal stop time to distinguish trajectories into stop episodes and move episodes. Speed threshold is dependent on the moving object and location where the object is moving. Research[14] extracted stops with user-specified minimum duration and diagonal length less than a user-specified distance threshold. Research[15] also used duration time but with an additional distance criteria for judging points in a stop location. Hybrid method could improve the accuracy to some extent, but it is hard to completely avoid the demerits mentioned above.

Overall, centroid-based method is not a good option in identifying stop without unknowing the number of stops. Due to the limitation of speed-based method, it can be utilized as an assisting variable to distinguish stops. Duration is a vital variable to identify stop from trajectories while it needs to be decided very carefully because of its sensitivity. Density-based methods use the spatial reflection of the relationship of activity and point density and require the data in a more frequent interval. Furthermore, density-based methods needed to consider the characteristics of trajectories when apply them. In this paper, we propose a two-step method which firstly uses a density-based method to identify all stops and then uses a supervised machine learning method to distinguish activity stops and non-activity stops. In the first step, constraints are included as improvements to make density-based methods to adjust to trajectory situations. In the second step, stop duration, size of the high-density cluster and distance to key locations are used to distinguish activity stop and non-activity stop.

3. DATA

The GPS data utilized in the research were collected by 30 volunteers in Nagoya area, Japan during 5 weeks in 2008. Each volunteer was assigned a mobile with GPS module which can record and send GPS information to the server every 10 seconds. However, sometimes the GPS module send GPS information longer than 10 seconds in case of tunnel, subway etc. Overall, 97.4% of the GPS intervals are less than 20 seconds. The GPS information sent back to server includes longitude, latitude, time stamp, signal quality etc. Respondents were required to mark the start, end, mode and purpose of each trip during the 5 weeks. Besides, socio-demographic information of each respondent was collected by questionnaires, including addresses of home and workplace, occupation, yearly income, driving license, daily primary transportation mode and so on. Fig. 1 demonstrates the basic aggregated statistic information of the dataset used in this research. Almost all volunteers are in age 20~65, which means work force age in Japan and almost all have a job or part time job, which means they are active trip maker. Auto, walking and rail are the main modes and business, back home and work are the main trip purpose in the dataset.

Total GPS trip data were almost equally divided into two datasets: dataset 1 for calibrating the key parameters in the algorithm; dataset2 for validating the algorithm with estimated parameters.



Note: one trip may contain more than one mode and it makes the total number of mode is bigger than total number of purpose

Fig. 1 Aggregated statistical results of dataset

4. METHODOLOGY

An improved DBSCAN algorithm used for distinguishing the stop points and move points from continuous GPS trajectories is advanced in this paper.

(1) Original DBSCAN algorithm

In this paper, we utilized the same notations as those given in research[16]. We apply the key definitions of DBSCAN in the context of GPS tracing points.

Definition 1: (Eps-neighborhood of a point) The Eps-neighborhood of a point, denoted by $N_{Eps}(p)$, is defined by $N_{Eps(p)}=\{q\in D|dist(p,q)\leq Eps\}$

Definition 2: (directly density-reachable) A point p is directly density-reachable from a point q wrt. Eps, MinPts if

1) $p\in N\ Eps(q)$ and
2) $|N\ Eps(q)| \geq MinPts$ (core point condition)

Definition 3: (density-reachable) A point p is density-reachable from a point q wrt. Eps and MinPts if there is a chain of points $p_1, \ldots, p_n$, $p_1=q$, $p_n=p$ such that p_{i+1} is directly density-reachable from p_i.

Definition 4: (density-connected) A point p is density-connected to a point q wrt. Eps and MinPts if there is a point o such that both, p and q are density-reachable from o wrt. Eps and MinPts.

Definition 5: (cluster) Let D be a database of points. A cluster C wrt. Eps and MinPts is a non-empty subset of D satisfying the following conditions:

1) $\forall p, q$: if $p \in C$ and q is density-reachable from p wrt. Eps and MinPts, then $q\in C$. (Maximality)
2) $\forall p, q \in C$: p is density-connected to q wrt Eps and MinPts. (Connectivity)

Definition 6: (noise) Let $C_1, \ldots, C_k$ be the clusters of the database D wrt. parameters Eps_i and $MinPts_i$, i=1, …, k. Then we define the nose as the set of points in the database D not belonging to any cluster C_i, i.e. noise = $\{p \in D|\forall i: p \notin C_i\}$

(2) Application to GPS points

When DBSCAN is applied in the situation of GPS track points, cluster is the equivalence of stop points which gather together with a higher density; noise is the equivalence of move points along links with a lower density.

DBSCAN algorithm was invented to solve the classification of spatial points without consideration of sequence among them. Consequently, in a detoured trajectory, one distinguished stop cluster may contain other move points or points in the subsequent clusters sharing the same location. Furthermore, due to the definitions and concepts in the original DBSCAN algorithm, points moving along a straight road with a low speed but high frequency of GPS signal transmission will be grouped into one cluster under certain given parameter values. As a result, applying original DBSCAN algorithm to GPS trajectories without any improvement will lead to mistake. Here, we advance ConstDBSCAN where two constraints are added to original DBSCAN and the ConstDBSCAN can overcome the demerit mentioned above.

(3) ConstDBSCAN

The first constraint is all points in a cluster should be temporally sequential. It means no "jump" of sequential order is allowed in the cluster. If this "jump" happens, the cluster will be divided into two potential clusters at the "jump point" and each one will be tested if it satisfies the condition of minimum number of points in one cluster. If so, the points in the potential cluster will be labeled as "noise", which means the move point. Otherwise the points in the potential cluster will be labeled as corresponding cluster number and tested by the second constraint below.

The second constraint is percentage (PCT) of abnormal points in a cluster should not exceed a given threshold named as PCT_{AP}. To be specific,

$$PCT \leq PCT_{AP} \qquad (1)$$

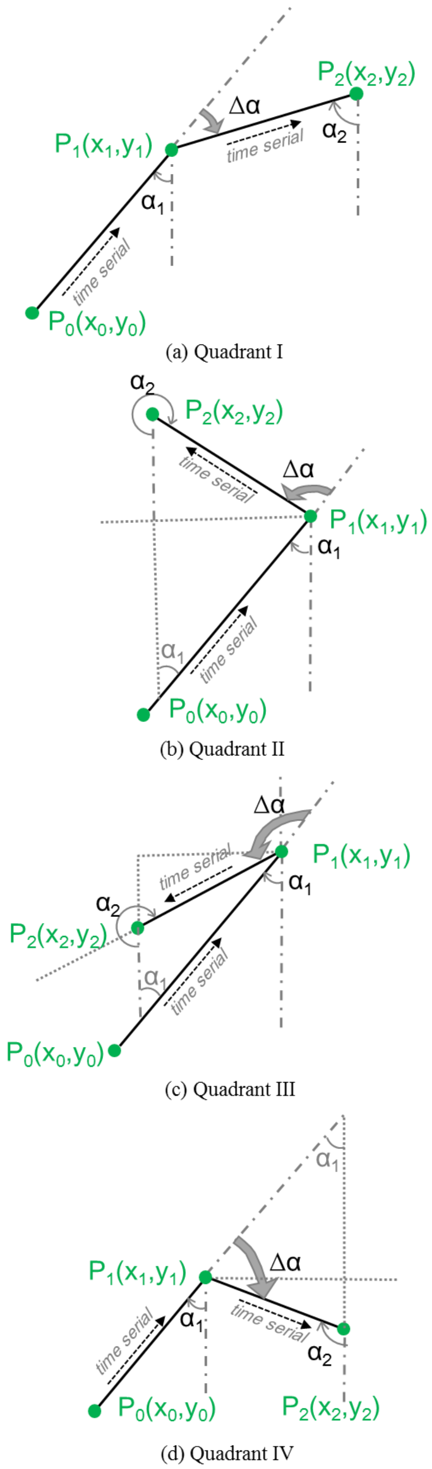
where $PCT = \frac{|Abnormal\ Points|}{|Cluster|}$, $|Abnormal\ Points|$ is the number of abnormal points in the cluster, and $|Cluster|$ is number of all points in the same cluster.

Before the definition of abnormal point is given, the direction and direction change of a point in a cluster should be given as follows. The direction of a point is defined in an imagined situation in a Cartesian coordinates where the point is origin and the direction is defined as the angle between the negative direction of vertical axis and the line between the point and its previous point, like α_1 for point P$_1$ and α_2 for point P$_2$ shown in Fig. 2. Suppose three points in the cluster are marked sequentially as P_0, P_1 and P_2. Direction change from point P_2 to point P_1 is defined as the angle from ray $\overline{P_0P_1}$ to ray $\overline{P_1P_2}$, shown as Δα in Fig. 2. Δα is equals to the angular difference between α_1 and α_2, i.e. $\Delta\alpha = \alpha_2 - \alpha_1$. Since we use the cosine value of Δα, it does not matter the Δα is a negative one or positive one.

In a cluster, if it is a stop location, the points in this location scatter and direction change of these points should not always be close to 0. It means that the cosine value of direction change (named as direction change coefficient, DCC) should not be always nearly 1. Points with a DCC value almost 1 means the respondent are moving along a link of a road. In a cluster, not all the DCC of all points should be nearly 1. So abnormal points are those points with a DCC close to 1. Here we use DCC_{AP} to denote the approximation to 1.

$$Abnormal\ Point = \{DCC \geq DCC_{AP} |\ Point \in Cluster\} \qquad (2)$$

The improved DBSCAN algorithm in this research, named as ConstDBSCAN, are shown in Fig. 3. Firstly, DBSCAN algorithm are applied to obtain the cluster points (stop points) and noise points (move points) in line 2. Then each cluster is tested by constraint 1. New cluster may split from the older one or the old cluster may be labeled as noise if it does not follow the rule of cluster. Finally the cluster satisfying constraint 1 will be tested by constraint 2. Clusters that satisfying constraint 1 and 2 are marked as stop points; other points will be marked as move points.



(a) Quadrant I

(b) Quadrant II

(c) Quadrant III

(d) Quadrant IV

Fig. 2 Direction of a point and Direction Change of two points when second point in different quadrants

5. RESULTS

This section explains the result of estimation of parameters and validation of applying ConstDBSCAN algorithm and estimated parameters. Dataset 1 is used for estimating the four parameters and dataset 2 is used for algorithm validation.

(1) Parameter estimation

In the ConstDBSCAN algorithm, there are 4 parameters needed to be estimated and they are the primary input variables. They are Eps, MinPts, DCC_{AP} and PCT_{AP}. Cumulative frequency method was used to estimate these 4 parameters. Estimation results of these four parameters are interpreted in Fig. 4.

Fig. 4-a shows that 95% of stop locations have a minimum number of point more than 4. If MinPts equals to 5, this percentage drops to 86.7% which is lower than the required 90%. Consequently, we use 4 as the estimated result of MinPts. This means that if MinPts equals to 4 points in the neighborhood, there is 90% probability a stop point is identified in clusters.

Fig. 4-b demonstrates that 90% of stop points have a distance less than 25 meters given MinPts equaling to 4. It means that if Eps equals to 25 meters, there is 90% probability that a stop point is identified in clusters.

DCC of abnormal points in the cluster can be estimated by obtaining DCC from move points. Fig. 4-c interpret that 90% move points have a direction change coefficient more than 0.8. It means that there is 90% probability that a point with DCC value more than 0.8 is a move point. If this point is in a cluster, it should be an abnormal point.

Fig. 4-d.1 shows that 86% of move point group between 2 clusters have a percentage of abnormal points more than 60% while Fig. 4-d.2 demonstrates that 93% of clusters have a percentage of abnormal points less than 60%. Consequently, the percentage of abnormal point is totally different in move points group and stop point cluster. So with a premise that DCC_{AP} equals to 0.8, PCT_{AP} equaling to 60% means if PCT_{AP} in a cluster candidate is less than 60%, there is 93% probability that this cluster candidate should be stop point cluster.

Finally we got the estimated parameters as follows: Eps = 25 meter, MinPts =4, DCC_{AP} =0.8 and PCT_{AP} =60%.

(2) Validation

An accuracy index should not only demonstrate the ability of accurately identify stop points but also move points. Consequently, an index named as IAISM (Index of Accurately Identifying Stop and Move points) is used to calculate the accuracy of the algorithm.

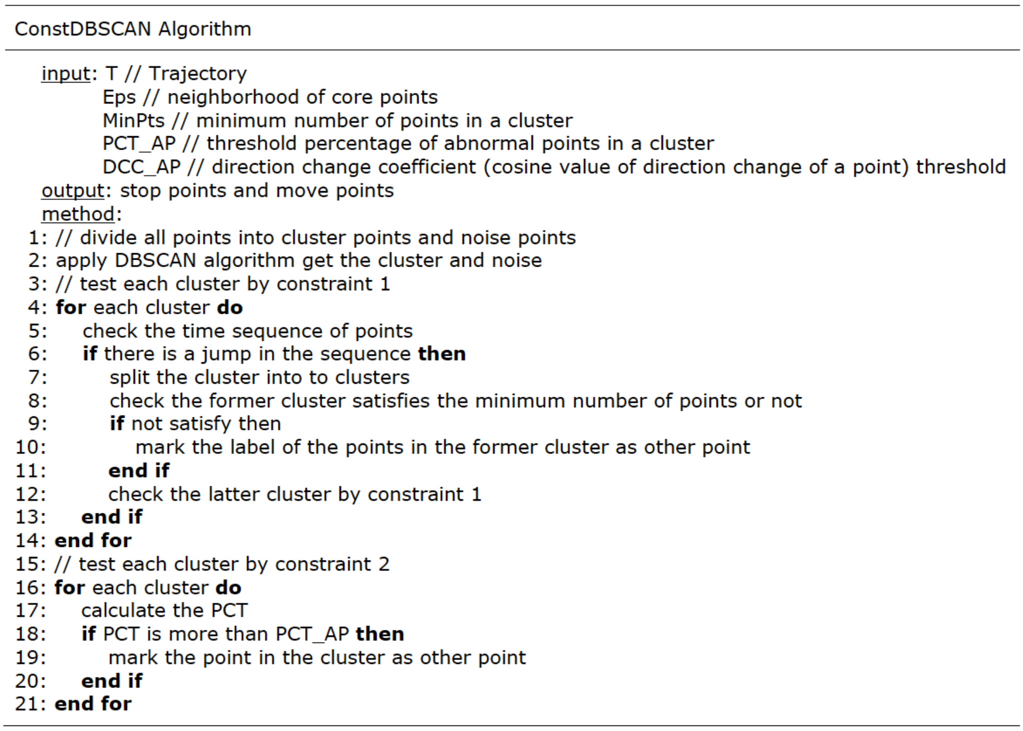
$$\text{IAISM} = \frac{N_{SS} + N_{MM}}{N_{SS} + N_{SM} + N_{MS} + N_{MM}} \quad (3)$$

Where N_{SS} is the number of stop points which are identified as stop points, N_{MM} is the number of move points which are identified as move points, N_{SM} is the number of stop points which are identified as move points and N_{MS} is the number of move points which are identified as stop points.

Dataset II is used to validate the algorithm and the estimated parameters. The average accuracy and minimum & maximum accuracy of dataset I and dataset II are shown in Table 2.

Table 2 Accuracy of two datasets

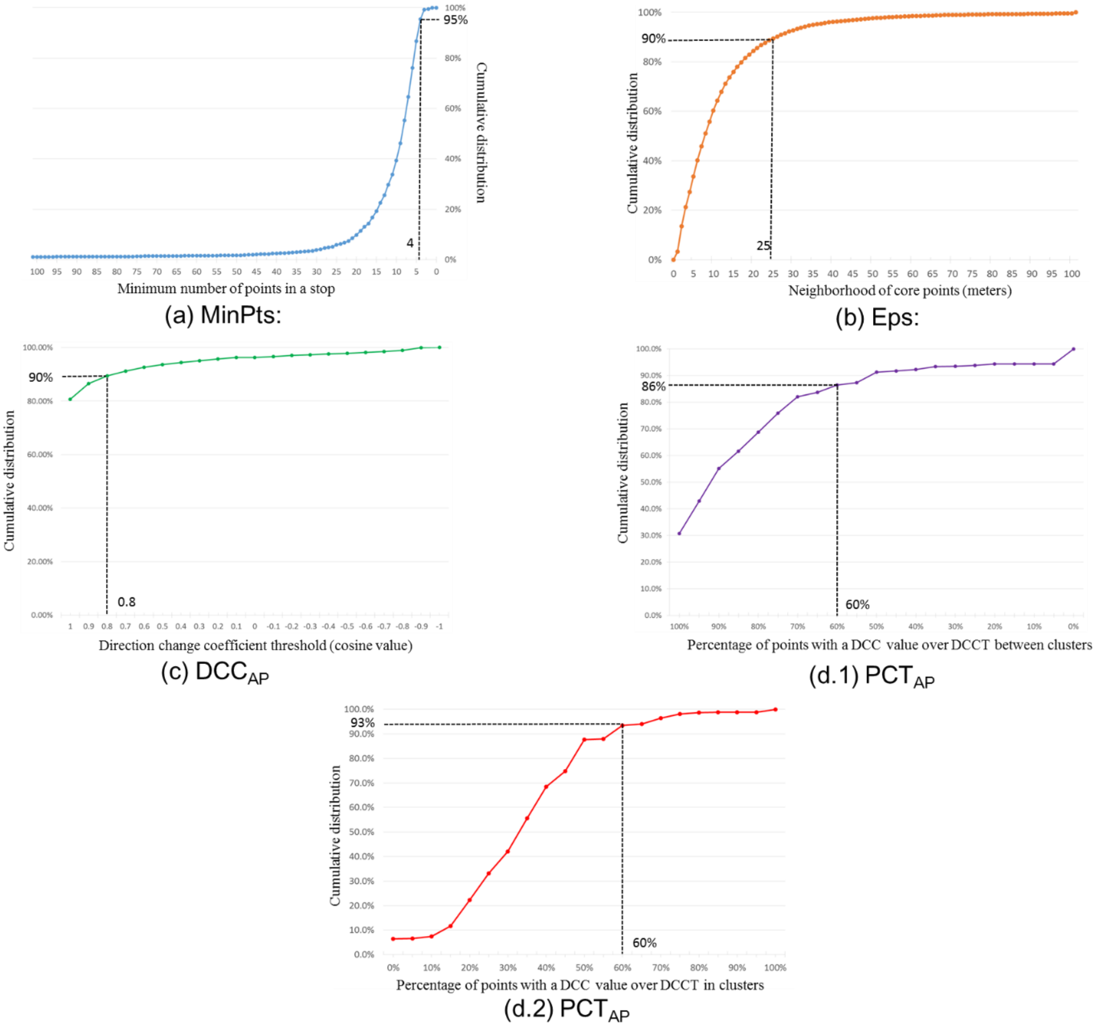
Data Set	**Average**	**Max**	**Min**
Data set I	89.6%	98.9%	57.3%
Data set II	90.7%	99.2%	42.6%



```
ConstDBSCAN Algorithm

input: T // Trajectory
       Eps // neighborhood of core points
       MinPts // minimum number of points in a cluster
       PCT_AP // threshold percentage of abnormal points in a cluster
       DCC_AP // direction change coefficient (cosine value of direction change of a point) threshold
output: stop points and move points
method:
// divide all points into cluster points and noise points
apply DBSCAN algorithm get the cluster and noise
// test each cluster by constraint 1
for each cluster do
    check the time sequence of points
    if there is a jump in the sequence then
        split the cluster into to clusters
        check the former cluster satisfies the minimum number of points or not
        if not satisfy then
            mark the label of the points in the former cluster as other point
        end if
        check the latter cluster by constraint 1
    end if
end for
// test each cluster by constraint 2
for each cluster do
    calculate the PCT
    if PCT is more than PCT_AP then
        mark the point in the cluster as other point
    end if
end for
```

Fig. 3 ConstDBSCAN algorithm



(a) MinPts:

(b) Eps:

(c) $\mathrm{DCC}_{\mathrm{AP}}$

(d.1) $\mathrm{PCT}_{\mathrm{AP}}$

(d.2) $\mathrm{PCT}_{\mathrm{AP}}$

Fig. 4 Estimation result

DJ-Cluster algorithm

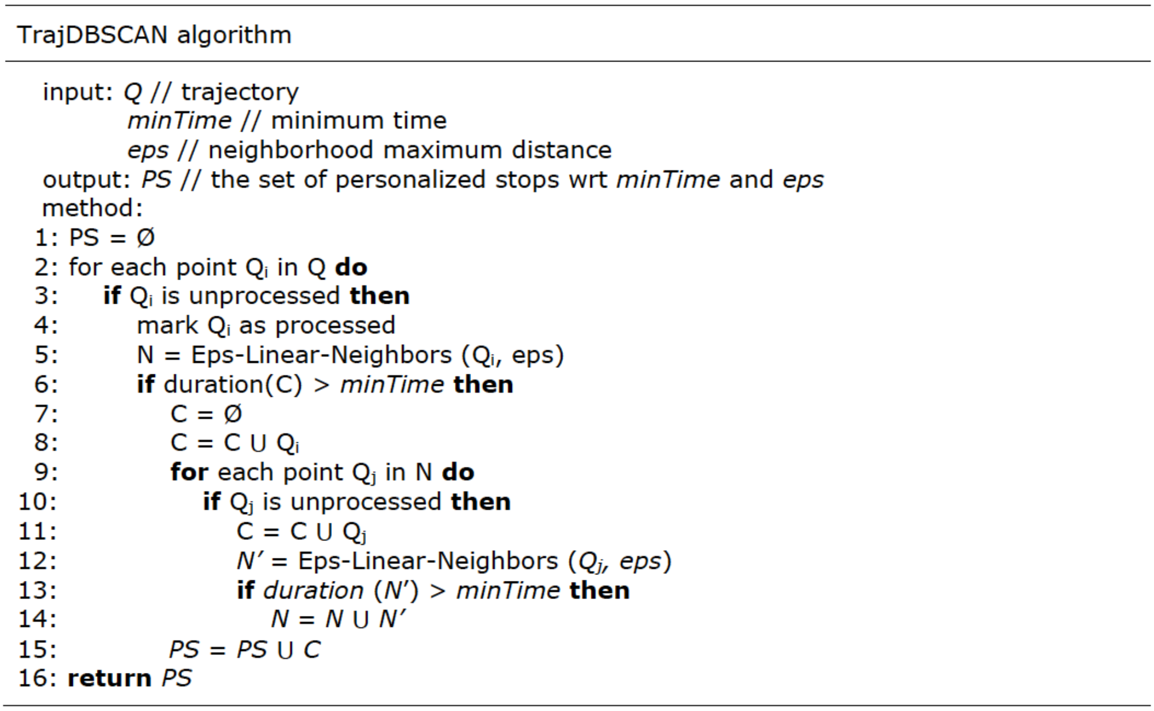
```
while there is at least one unprocessed point $p$ in sample $S$ do
    compute the density-based neighborhood $N(p)$ wrt $Eps$ and $MinPts$
    if $N(p)$ is null ($p$ is not in a cluster) then
        label $p$ as noise
    else if $N(p)$ is density-joinable to an existing cluster then
        merge $N(p)$ and its all density-joinable clusters
    else
        create a new cluster C based on $N(p)$
    end if
end while
```

Fig. 5 DJ-Cluster algorithm

CB-SMoT algorithm

```
Input: $T$ // set of trajectories
       $A$ // application
       $a$ // area for quantile function
       $minTime$ // minimum time for clustering
Output: $S$ // set of stops
        $M$ // set of moves
Method:
For each trajectory $t$ in $T$ do
 // compute the cluster
    set clusters as empty
    $Eps$ = quantile ( $\mu(t)$, $\sigma(t)$, $a$)
    for each unprocessed point $p$ in $t$ do
        neighbors = linear_neighborhood ($p$, $Eps$)
        if $p$ is a core point wrt $minTime$, $Eps$
            for each neighbor $n$ in neighbors do
                add to neighbors every unprocessed point in linear_neighborhood ($n$, $Eps$)
                set neighbors as a cluster in clusters
                set all points in neighbors as processed
            end for
    end for
// find stop sand moves
    for each cluster in clusters do
        for each intersection with a different $Rc$ with duration time $t \leq \Delta C$ do
            generate a stop in $S$
        end for
        for each subtrajectory that is not stop do
            if duration $\geq$ $minTime$
                generate an unknown stop in $S$
        end for
    end for
    for each subtrajectory which is not a stop do
        generate a move in $M$
    end for
end for
```

Fig. 6 CB-SMoT algorithm



```
TrajDBSCAN algorithm

input: Q // trajectory
       minTime // minimum time
       eps // neighborhood maximum distance
output: PS // the set of personalized stops wrt minTime and eps
method:
PS = Ø
for each point $Q_i$ in Q do
    if $Q_i$ is unprocessed then
        mark $Q_i$ as processed
        N = Eps-Linear-Neighbors ($Q_i$, eps)
        if duration(C) > minTime then
            C = Ø
            C = C U $Q_i$
            for each point $Q_j$ in N do
                if $Q_j$ is unprocessed then
                    C = C U $Q_j$
                    N′ = Eps-Linear-Neighbors ($Q_j$, eps)
                    if duration (N′) > minTime then
                        N = N U N′
            PS = PS U C
return PS
```

Fig. 7 TrajDBSCAN algorithm

Table 3 Parameter value used in variations of DBSCAN and accuracy comparison

Algorithm	Parameter value						Accuracy
	Min duration (seconds)	Min number of points	Neighborhood (meters)	Area	DCC_{AP}	PCT_{AP}	
DJ-cluster	---	10	20	---	---	---	78%
DB-SMoT	120 or 300	---	---	0.46	---	---	54%
TrajDBSCAN	180	---	30	---	---	---	69%
ConstDBSCAN	---	4	25	---	0.8	60%	90%

6. COMPARISON

In this section, ConstDBSCAN and other variants of DBSCAN algorithms are compared.

ConstDBSCAN algorithm advanced in this paper are compared with 3 other variations of DBSCAN algorithm in this subsection. Since in the corresponding papers, no accuracy was included, these algorithms are tested by our data set to obtain the accuracy in order to make comparisons.

The first one is called DJ-cluster algorithm[12), a simplified version of DBSCAN algorithm. It uses the same concept of core point as in DBSCAN. However, as far as the principle of expand the cluster is concerned, density-reachability and density-connectivity are replaced by density-joinability. Instead of using core points for cluster's expansion, any sharing point in any two clusters can be joined together as one cluster in this algorithm (Fig. 5).

The second one is called CB-SMoT algorithm[7) which modified some concepts in the original DBSCAN algorithm. Instead of using straight distance in the original DBSCAN algorithm, CB-SMoT uses the distance along the trajectory. Besides, it replaced the minimum number of points by the minimum stop duration inside the neighborhood of a core point. Furthermore, it advanced a quantile function to calculate a parameter, named area. Then the parameter Eps can be calculated with the information of approximate proportion of points that generate potential stops in relation to the total amount of points in the trajectory. It is shown in Fig. 6.

The third one is called TrajDBSCAN algorithm[6). Compared to the original DBSCAN algorithm, TrajDBSCAN uses temporal linear neighborhood in which minimum stop duration takes place of minimum number of points in a neighborhood to be used as a key feature of core points (Fig. 7).

Table 3 shows utilized parameters for testing and accuracy results. Parameters are recommended values or calculated following the methods in their papers. It shows that the ConstDBSCAN has a higher accuracy compared to other variants of DBSCAN.

7. CONCLUSIONS

In this paper, we advanced a ConstDBSCAN algorithm for identifying stop points from a series of GPS track points. Two constraints, sequence constraint and direction change constraint are applied to DBSCAN algorithm. After comparing it to other variants of DBSCAN algorithm, it is shown that ConstDBSCAN achieved a higher accuracy to other 3 variants.

REFERENCES

1) Gong, L., Morikawa, T., Yamamoto, T., & Sato, H. (2014). Deriving Personal Trip Data from GPS Data: A Literature Review on the Existing Methodologies. *Procedia-Social and Behavioral Sciences,* 138, 557-565.
2) Ashbrook, Daniel, and Thad Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing 7.5 (2003)*: 275-286.
3) Agamennoni, G., Nieto, J., & Nebot, E. (2009, May). Mining GPS data for extracting significant places. In Robotics and Automation, 2009. *ICRA'09. IEEE International Conference on (pp. 855-862).* IEEE.
4) Mizuno K., Kanamori R., Sano S., Nakajima S. and Ito T., Identifying move and stop in GPS data with Support Vector Machines, *Conference of Infrastructure Planning and Management , JSCE,* 2013 (in Japanese)
5) Cao, Xin, Gao Cong, and Christian S. Jensen. Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment 3*, no. 1-2 (2010): 1009-1020.
6) Tran, Le Hung, et al. Robust and Hierarchical Stop Discovery in Sparse and Diverse Trajectories. *No. EPFL-REPORT-175473*. 2011.
7) A. Tietbohl Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares. A clustering-based approach for discovering interesting places in trajectories. *SAC,* 2008.
8) Xie, Kexin, Ke Deng, and Xiaofang Zhou. From trajectories to activities: a spatio-temporal join approach. In *Proceedings of the 2009 International Workshop on Location Based Social Networks*, pp. 25-32. ACM, 2009.
9) Alvares, Luis Otavio, Vania Bogorny, Bart Kuijpers, Jose Antonio Fernandes de Macedo, Bart Moelans, and Alejandro Vaisman. A model for enriching trajectories with semantic geographical information. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, p. 22. ACM, 2007.
10) Zimmermann, Max, Thomas Kirste, and Myra Spiliopoulou. Finding stops in error-prone trajectories of moving objects with time-based clustering. In *Intelligent Interactive Assistance and Mobile Multimedia Computing*, pp. 275-286. Springer Berlin Heidelberg, 2009.
11) Kami, Nobuharu, Nobuyuki Enomoto, Teruyuki Baba, and Takashi Yoshikawa. Algorithm for detecting significant locations from raw GPS data. In *Discovery Science*, pp. 221-235. Springer Berlin Heidelberg, 2010.
12) Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, and Loren Terveen. Discovering personally meaningful places: An interactive clustering approach. *ACM Transactions on Information Systems,* 25(3):56–68, July 2007
13) Yan, Zhixian, Christine Parent, Stefano Spaccapietra, and Dipanjan Chakraborty. A hybrid model and computing platform for spatio-semantic trajectories. In *The Semantic Web: Research and Applications,* pp. 60-75. Springer Berlin Heidelberg, 2010.
14) Andrienko, Gennady, Natalia Andrienko, Georg Fuchs, Ana-Maria Olteanu Raimond, Juergen Symanzik, and Cezary Ziemlicki. Extracting semantics of individual places from movement data by analyzing temporal patterns of visits. In *Proceedings of The First ACM SIGSPATIAL International Workshop on Computational Models of Place (COMP'13)*. 2013.
15) Leclerc, B., Trépanier, M., & Morency, C. (2013). Unraveling the Travel Behavior of Carsharing Members from Global Positioning System Traces. *Transportation Research Record: Journal of the Transportation Research Board,* 2359(1), 59-67.
16) Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd, vol. 96,* pp. 226-231. 1996..

(Received August 1, 2014)