GPS Trajectory Clustering and Visualization Analysis

Article *in* Annals of Data Science · March 2018

DOI: 10.1007/s40745-017-0131-2

4 authors, including:

Li Cai
Fudan University
16 PUBLICATIONS **486** CITATIONS

SEE PROFILE

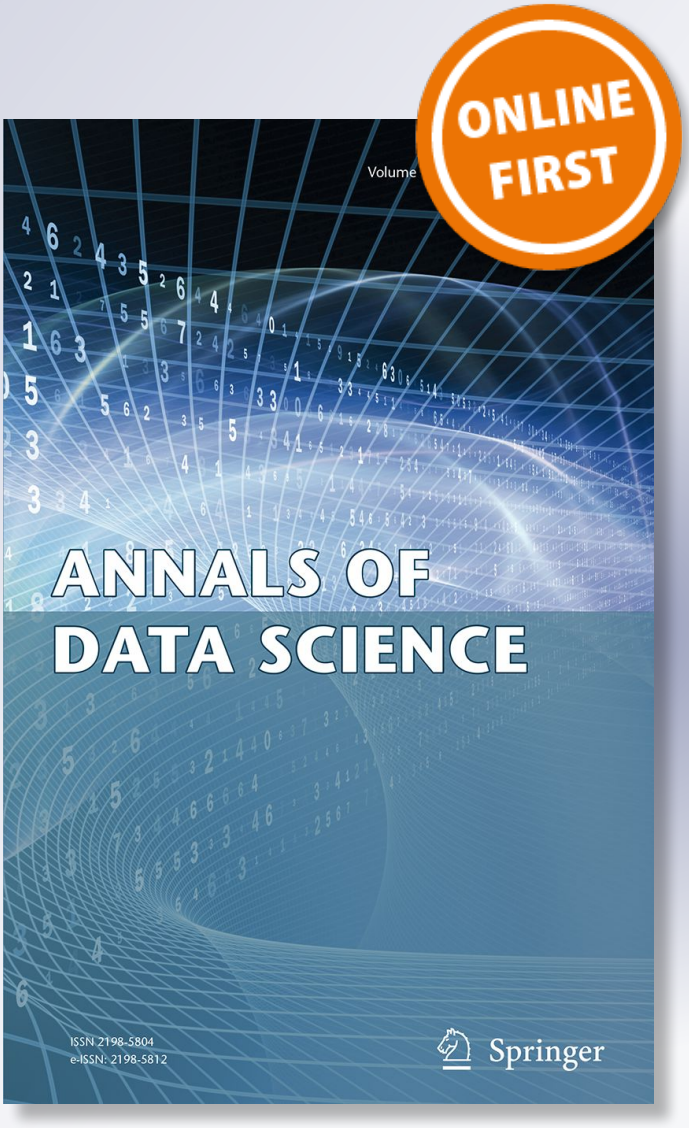
GPS Trajectory Clustering and Visualization Analysis

Li Cai, Sijin Li, Shipu Wang & Yu Liang

Annals of Data Science

ISSN 2198-5804

Ann. Data. Sci.
DOI 10.1007/s40745-017-0131-2



Springer

Ann. Data. Sci.
https://doi.org/10.1007/s40745-017-0131-2

GPS Trajectory Clustering and Visualization Analysis

Li Cai[1,2] · **Sijin Li**[2] · **Shipu Wang**[2] · **Yu Liang**[2]

Received: 15 October 2017 / Revised: 17 October 2017 / Accepted: 21 October 2017

Abstract The trajectory data of taxies containing time dimensional and spatial dimensional information is an important kind of traffic data. How to obtain valuable information from these data has become a hot topic in the field of intelligent transportation. Existing trajectory clustering algorithms can only compute similarities using partial characteristics of the trajectory data, leading to clustering results are not accurate. This study proposes a novel trajectory clustering algorithm named GLTC, which can obtain more accurate number of clusters based on the global and local characteristics of trajectories. This study intuitively displays the laws and knowledge in clustering results using visualization techniques. Experimental results reveal that the GLTC algorithm can discover more accurate clustering results, effectively display spatial-temporal change trends in GPS data, and better assist in analyzing the flow law of urban citizens and urban traffic conditions using visualization methods.

Keywords Trajectory clustering · GPS trajectory data · Visualization · Global characteristics · Local characteristics

✉ Li Cai
lcai@fudan.edu.cn

Sijin Li
sijinmj@ynu.edu.cn

Shipu Wang
spwang@ynu.edu.cn

Yu Liang
yuliang@ynu.edu.cn

1 School of Computer and Science, Fudan University, Shanghai 200433, China

2 School of Software, Yunnan University, Kunming 650091, China

Published online: 21 November 2017

1 Introduction

With the wide applications of mobile positioning system, such as intelligent terminal, wireless communication, GPS device, it becomes easier to get the location information of moving objects [1]. Because the trajectory data contains a lot of information, researchers can obtain the motion pattern of the moving objects, identify the popular routes and mining the spatio-temporal characteristics of the residents' travel by clustering analysis. The research results can be used in urban traffic management, road planning and location-based services (LBS) and so on [2].

In recent years, the research on trajectory clustering has been widely concerned by scholars. Zheng [3] through mining the history of tens of thousands of taxi trajectory in Beijing, based on the taxi driver on the city road rich knowledge and experience, the use of road network data for the drivers personalized recommendation of fast driving routes. Davies et al. [4] proposed a method of constructing urban road network, using trajectory sampling point to gradually improve the existing main road network, this method can identify and build a simple road shape and trend direction. Camargo et al. [5] used vector lines to express the complete trajectory, through historical data to establish regression models, calculate the similarity between trajectories and models, and then clustering to get the motion pattern. Kharrat et al. [6] proposed NETSCAN, a density-based trajectory clustering method, which first calculates the busy path based on the path through which the moving object passes, and then clusters the sub-tracks according to the density parameters set by the user. Roh and Hwang [7] proposed a new distance metric based on the Hausdorff distance. And on this basis, put forward an effective clustering algorithm NNCkister, the main idea of this algorithm is that the two similar trajectories have the same nearest neighbor.

It can be seen from the above studies that many scholars have proposed a number of trajectory clustering algorithms, but these algorithms have some limitations as follows:

1. Existing trajectory clustering algorithms usually regard the entire trajectory or segmented trajectory as a basic unit involved in clustering calculation. These methods not only easily ignore the consistency of the internal structure of the trajectory, resulting in could not discover the common sub-patterns in the trajectory, but also overlook the whole feature of the trajectory, causing lower accuracy of clustering results.
2. Lacking of intuitive, fine-grained visualization display modes, and cannot visually explain the hidden knowledge and regularity existing in clustering results, resulting in poor readability of the results.

To solve the above problems, this paper presents a novel clustering algorithm for GPS trajectory data, named Global and Local Trajectory Clustering algorithm, abbreviated as GLTC, which has the following contributions:

1. Considering global characteristics of the trajectory, we can obtain the feature information of the whole trajectory. Then, utilizing local characteristics of the trajectory, the trajectory is divided into different sub-modes, which avoids the non-uniform structure in the trajectory. We use the complementary relationship between two

trajectory characteristics to improve the accuracy and effectiveness of clustering results, and mine more laws accord with the residents' daily traveling.
2. Some visual techniques are used to display the trajectory clustering results, for example, different colors can identify the trajectory source and destination and the results of global clustering and local clustering. In this way, we avoid visual confusion caused by massive data, and help researchers to better discover the mobility laws of the residents.

2 Literature Review

Some research scholars have done a lot of research work based on the characteristics of the trajectory itself in order to discover the hidden and unknown knowledge in the data, so as to improve the accuracy and interpretability of the clustering results. Vlachos et al. [8] proposed a non-metric similarity function based on the longest common sub-sequence (LCSS) to improve the similarity between trajectories by increasing the weight of similar parts of the sequence and using unsupervised learning to achieve trajectory clustering. Pelekis et al. [9] proposed a trajectory clustering method for GenL IP, which is multiplying a series of polygonal areas of the two trajectories by the corresponding weights to obtain the values as the distance between the two trajectories. Shen et al. [10] proposed a density-based ε-distance trajectory clustering algorithm, which was based on the similarity of the start-end points of the trajectory to partition the trajectory set and used the density-based method to achieve trajectory clustering. Yuan et al. [11] proposed a trajectory clustering algorithm based on structural similarity. The trajectory was partitioned into a trajectory segments, and the matching degree of the trajectory was judged by the structural similarity of the trajectory. Lee et al. [12] proposed a density-based trajectory segmentation clustering method. Each trajectory is divided into trajectory segments by the two trajectories at the furthest distance and the same driving direction. The distance between the two trajectory segments is calculated as the measure of the similar trajectory segments, and a similar trajectories are obtained by the density-based method. Buchin et al. [13] proposed a general framework for trajectory segmentation based on space-time rules. This method optimally divided the trajectories according to the given spatio-temporal rules, so that the number of trajectories after division is the smallest and the spatio-temporal features in each trajectory segment are kept uniform.

The above methods for trajectory clustering are mainly divided into two types: clustering algorithm based on the whole trajectory similarity and clustering algorithm based on the partial trajectory segment similarity. In the literature [8–10,14], the clustering methods based on the whole trajectory similarity are adopted, their core ideas are to take the whole trajectory as the basic unit, and calculate the similarity between any two trajectories as the foundation of trajectory clustering, this way can grasp the scope of the hot path as a whole. In [11–13,15], clustering methods based on the similarity degree of the trajectory segment are adopted, these methods divide the whole trajectory into several trajectory segments according to the feature points, and execute to cluster based on the similarity between the trajectory segments, it can solve the problem of comparison between complex trajectories.

Table 1 Notations

Notation	Description
TD	Trajectory set, $TD = \{TD_1, TD_2, \ldots, TD_i, \ldots, TD_m\}$
TS	Trajectory segment set, $TS = \{L_1, L_2, \ldots, L_i, \ldots, L_n\}$
$p(L_i)$	The number of sampling points in Li trajectory segment
$\theta^k{}_S, \theta^k{}_E$	The direction angle of start and end sampling points in TD_k trajectory
$\theta^k{}_{c1}, \theta^k{}_{c2}$	The direction angle of middle sampling points in TD_k trajectory, when the number of sampling points is odd, θ^k_{c1} is the same as θ^k_{c2}, for even, θ^k_{c2} is the direction angle of θ^k_{c1} after sampling points
D_S, D_{Mi}, D_E	The euclidean distance of start point, midpoint and end point of between two trajectories, $i = 1$ or 2
$D_{Thred}, S_{Thred}, T_{Thred}$	The threshold of the similar trajectory distance, starting distance, and the number of trajectory clustering
$sim, \omega, delta$	Similarity, Corner and Neighbor Set Thresholds

3 Overview of the GLTC Algorithm

This section defines the terminologies, principle and implementation method of the GLTC algorithm.

3.1 Principle of GLTC Algorithm

The GLTC algorithm consists of two stages, the first stage: based on global characteristics of trajectory, performing clustering for the trajectory; the second stage: based on local characteristics of trajectory, performing clustering for the trajectory segments, then integrating the results of two stages to complete the entire clustering process. Here, we define the notations (in Table 1) used in this paper.

3.2 Definitions of GLTC Algorithm

1. Trajectory clustering

In the first stage of the GLTC algorithm, the global features means the whole distance (D) between two trajectories [10]. Firstly, we calculate D, then D is used as the standard to measure the similarity between the trajectories, which including two distance thresholds S_{Thred} and D_{Thred}. From Fig. 1, we can see the relationship between the trajectories TD_i and TD_k. In order to describe their relationship, this section gives the following definition:

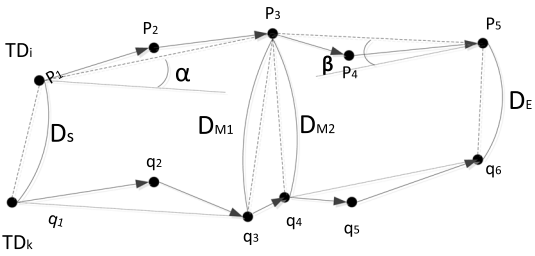


Fig. 1 The relationship between two trajectories

Definition 1 Middle distance $d(TD_{ic}, TD_{kc})$. The midpoint distance between TD_i and TD_k, whose calculation is divided into two cases, as shown in Eq. 1.

$$d(TD_{ic}, TD_{kc}) = \begin{cases} D_{M1} & TD_i \text{ and } TD_k \text{ both have odd number of points} \\ \frac{1}{2}D_{M1} + \frac{1}{2}D_{M2} & TD_i \text{ or } TD_k \text{ has even number of points} \end{cases} \tag{1}$$

Definition 2 The whole distance between trajectories $D(TD_i, TD_k)$. This distance consists of three parts: the starting point distance between trajectories, the midpoint distance and the end point distance, and can be calculated with Eq. 2. The weight values of W_S and W_E are determined by Eqs. 3 and 4 and W_M is set to 1/3.

$$D(TD_i, TD_k) = W_S D_S + W_E D_E + W_M d(TD_{ic}, TD_{kc}) \tag{2}$$

$$W_S = \begin{cases} \frac{1}{3}\left[1 - \frac{1}{2}sin\alpha - \frac{1}{2}sin\left(|\alpha - \beta|\right)\right] & \alpha \in [0, \frac{\pi}{2}) \\ \frac{1}{3}\left[1 + \frac{1}{2}sin\alpha + \frac{1}{2}sin(|\alpha - \beta|)\right] & \alpha \in [\frac{\pi}{2}, \pi) \end{cases} \tag{3}$$

$$W_E = \begin{cases} \frac{1}{3}\left[2 + \frac{1}{2}sin\alpha + \frac{1}{2}sin(|\alpha - \beta|)\right] & \alpha \in [0, \frac{\pi}{2}) \\ \frac{1}{3}\left[2 - \frac{1}{2}sin\alpha - \frac{1}{2}sin(|\alpha - \beta|)\right] & \alpha \in [\frac{\pi}{2}, \pi) \end{cases} \tag{4}$$

where α is the angle between two start-midpoint dashed line segments of TD_i and TD_k, β is the angle between two midpoint-endpoint dashed line segments, which can be calculated with Eqs. 5 and 6.

$$\alpha = |(\theta_{c1}^k - \theta_S^k) - (\theta_{c1}^i - \theta_S^i)| \tag{5}$$

$$\beta = |(\theta_E^k - \theta_{c2}^k) - (\theta_E^i - \theta_{c2}^i)| \tag{6}$$

2. Trajectory segments clustering

The second stage of the GLTC algorithm is based on the first stage, we add three local features of direction, corner and position, and use DBSCAN clustering algorithm [16] to realize clustering for trajectory segments [11]. We first calculate the corner of the trajectory to divide the trajectory segment, combining with the local features to calculate the structural distance between the trajectory segments. Then the distance is used as the standard to measure the similarity between the trajectory segments. Here, three parameters *delta*, *sim* and ω need to specify the appropriate value for completing the trajectory segments clustering to acquire the neighborhood of the trajectory segment.

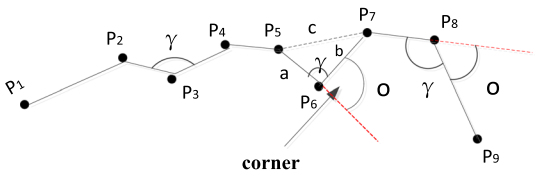


Fig. 2 The corner and included angle of trajectories

Definition 3 Trajectory corner (o). The corner between adjacent sampling points in the trajectory reflects the movement of the trajectory. As shown in Fig. 2. We can calculate the corner o by the angle γ, which can be calculated with Eqs. 7 and 8. From Fig. 2a, b are adjacent sides of the angle γ, c is opposite side of the angle γ, then according to ω, the sampling points meeting the condition of $|o| > \omega$ are partitioned.

$$\gamma = arccos((a^2 + b^2 - c^2)/2ab) \tag{7}$$

$$o = \begin{cases} \pi - \gamma & |\vec{a} \times \vec{b}| \geq 0 \\ \gamma - \pi & |\vec{a} \times \vec{b}| < 0 \end{cases} \tag{8}$$

Definition 4 Structural similarity $Ssim(L_i, L_j)$. The similarity between trajectory segments is measured by calculating the structural distance ($SDist(L_i, L_j)$) between trajectory segments that do not belong to the same trajectory, where W_D, W_A and W_L correspond to the weights of the different features(each weight value is greater than or equal to zero and their sum is equal to one). Finally, we do the normalized ($Norm()$) of the structural distance to convenience comparison.

$$SDist(L_i, L_j) = W_D Dir(L_i, L_j) + W_A Angle(L_i, L_j) + W_L Loc(L_i, L_j) \tag{9}$$

$$Ssim(L_i, L_j) = 1 - Norm(SDist(L_i, L_j)) \tag{10}$$

Definition 5 Direction $Dir(L_i, L_j)$. Direction represents the degree of deviation of the trajectory segments L_i and L_j belonging to the different trajectories in the moving direction. Where Φ is the angle of start-end dashed line segments between two trajectories.

$$Dir(L_i, L_j) = \begin{cases} min(||L_i||, ||L_j||) \times sin(\phi) & 0 \leq \phi \leq 90 \\ min(||L_i||, ||L_j||) & 90 \leq \phi \leq 180 \end{cases} \tag{11}$$

Definition 6 Corner $Angle(L_i, L_j)$. Corner represents the change in direction and the degree of fluctuation within the trajectory segments.

$$Angle(L_i, L_j) = \frac{\sum_1^{min(P(L_i), P(L_j))} ((|O_i - O_j|)/(|O_i| + |O_j|))}{P(L_i) + P(L_j)} \tag{12}$$

Definition 7 Position $Loc(L_i, L_j)$. Position represents the positional distance of the trajectory segments L_i and L_j, which uses the Hausdorff distance and can be calculated with Eqs. 13 and 14.

Table 2 Pseudocode of the GLTC algorithm

```
Algorithm: Global and Local Trajectory Clustering (GLTC)
Input: TD, S_Thred, D_Thred, T_Thred                    // The first stage
Output: G_TD ={ TD_1, TD_2, ……, TD_n}    (The result set of Trajectory Clustering)
While (TD.size()!=0)
    RL={}
    First=TD.getfirst()    /* get the first trajectory record from the TD set*/
    RL.add(First)
    for each TD_i in TD\{ First } do
        D=distance(First, TD_i)
        if D<D_Thred && D_S<S_Thred then
          RL.add(TD_i)     /* add the result in the RL set*/
        end if
    end for
    if RL.nums≥T_Thred  then
        G_TD.add(RL)
     end if
    TD.remove(First)
end
Input: G_TD, delta, sim, ω                    // The second stage
Output: L_TS={C_1, C_2, ……, C_n}(The result set of Trajectory segments Clustering)
/*The first step: Partitioned trajectories into trajectory segments */
for each TD_i ∈G_TD do
       Compute_Angle(TD_i)
       TS←according to ω Partitioned TD_i
end
/*The second step: computed the similarity degree between trajectory segment*/
length=TS.length
ssim=Matrix(length, length)
for each L_i, L_j∈TS∧i≠j∧L_i .parentid ≠ L_j .parentid do
       ssim[i][j]=Ssim(L_i, L_j, W)    /*Calculate the similarity of the trajectory segments
                                          by weight*/
       if ssim[i][j]≥sim then
            N(L_i)←L_j   /*Add L_j in the Neighbor Set N of L_i*/
        end if
end
/*The third step: Implementation of trajectory Clustering */
L_TS=Dbscan(TS, N, delta)
end
```

$$Loc(L_i, L_j) = max(h(L_i, L_j), h(L_i, L_j)) \tag{13}$$

$$h(L_i, L_j) = max(min(dist(p(L_i), p(L_j)))) \tag{14}$$

Definition 8 Neighbor Set N. For the trajectory segment L_i, if there is a trajectory segment $L_j (i \neq j)$ which satisfies $Ssim(L_i, L_j) \geq sim$, then L_j belongs to the neighbor set of L_i.

With the above definition, the GLTC algorithm takes into account the global and local features of the trajectory, so it improves the reliability and credibility of the clustering results. The algorithm's pseudocode is described in Table 2.

Here, we analyze the time complexity of GLTC algorithm. Firstly, $D(TD_i, TD_k)$ and D_S are calculated and compared with D_{Thred} and S_{Thred}, the trajectory satisfying the condition is added to the RL, and then the trajectory clustering is completed

according to T_{Thred}, and its time complexity is $O(n^2)$. Secondly, trajectories first are partitioned into trajectory segments, and the time complexity is $O(n)$; then, computing the similarity degree between trajectory segments, and the time complexity is $O(n^2)$; finally, implementation the trajectory clustering, and the time complexity is $O(n^2)$. Therefore, the total time complexity of this algorithm is $O(n^2)$.

4 Experiments

4.1 Data Source and Experimental Environment

GLTC algorithm is implemented by Java and Python languages and runs on a normal PC (Inter core i5 CPU, 16G memory, the development environment is MyEclipse10). Trajectory dataset was collected from 6599 taxies in Kunming. The time is from August 13 to 19, 2012, including the weekdays and weekends. To evaluate the trajectory dataset, we choose three representative time slot: 8: 00–9:00, 17:00–18:00 and 21:00–22:00 from weekdays, and 10:00–11:00, 19:00–20:00 and 22:00–23:00 from weekends.

4.2 GLTC Algorithm Implementation

GLTC algorithm needs to use different parameters, the selection process of various parameters is described in the following.

- Determination of global parameters

We calculate the clustering results under different parameters ($S_{Thred} = 1000$, 2000, 3000, 4000, 5000). The selection of the S_{Thred} needs to be considered with the D_{Thred} threshold. From the Fig. 3 we can see, when the threshold of S_{Thred} is 1000, 2000 and 3000, the change of the curve is very significant. When the threshold of S_{Thred} is taken to 4000, the trend tends to be stable. If the value of S_{Thred} is set too small, it will lead to the number of clusters being relatively small, because the clustering results tend to consider only the global characteristics of the trajectory, while ignoring the local similarity of the trajectory. Therefore, when $S_{Thred} = 3000$m and $D_{Thred} = 800$m, the effect of the trajectory clustering is the best.

- Determination of local parameters

Figure 4 shows the local clustering results under different threshold *delta* at different time slot. When *delta is* 6, the number of clusters is relatively large, leading to trajectories belonging to a cluster are divided into different clusters; when *delta* = 15, the number of clusters is relatively small, leading to trajectories belonging to the different clusters are divided into a cluster. Therefore, when *delta* is 10, the clustering effect is the best.

Besides the *delta* parameter, the different values of parameter *sim* also have a certain influence on the number of clusters. When the values of *sim* are 0.97 or 0.99, the number of clusters is very small. But when *sim* is 0.95, the clustering result has a

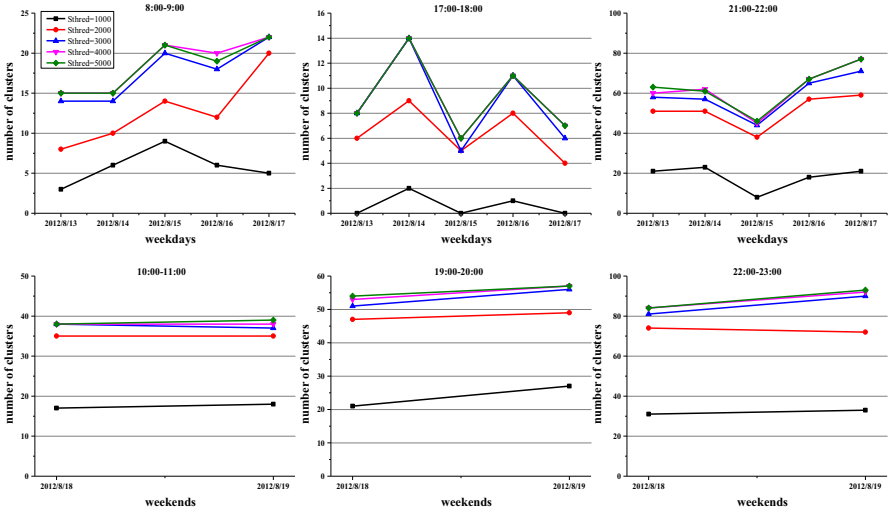


Fig. 3 Comparison of clustering results under different dates (weekdays and weekends) and different threshold

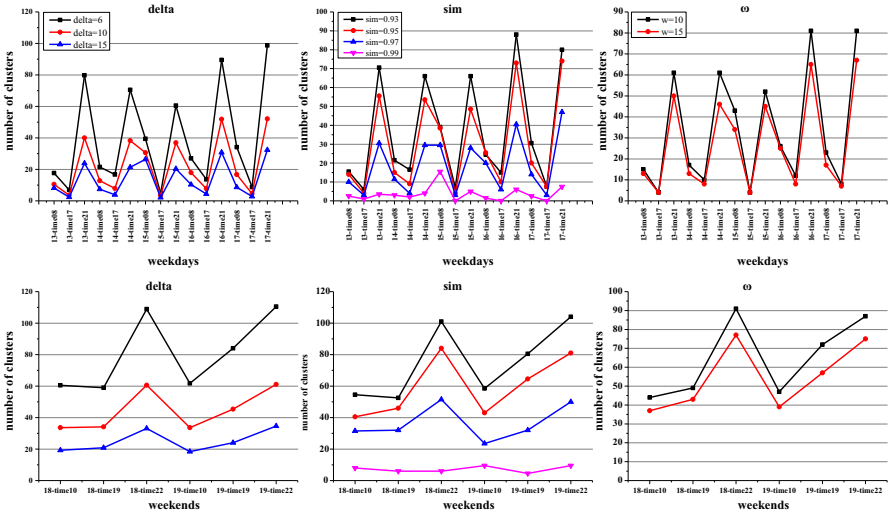


Fig. 4 Comparison of clustering results under different dates and thresholds

great difference, and the maximum difference of clustering number is close to 50, the reason for this difference is that the similarity is too high, resulting in the trajectory belonging to a cluster does not formed, so the threshold *sim* is 0.95. Moreover, the threshold ω has a certain influence on clustering results. When $\omega = 10$, the results are better than the clustering results when ω being 15, therefore the threshold ω of this paper is 10.

Table 3 Total clustering results of GLTC algorithm

Category	Date	Number of clusters
Weekdays	2012/8/13	80
	2012/8/14	88
	2012/8/15	99
	2012/8/16	119
	2012/8/17	112
Weekends	2012/8/18	184
	2012/8/19	206

4.3 Visualization Analysis of Trajectory Clustering Results

The emergence of visualization provides a new approach and method for trajectory data analysis and result validation, and makes up for computer's weakness in data analyses [17–19]. We combine the ArcGIS software with the Python language to present the clustering results of the GLTC algorithm by different visual means in the map. The heat of the popular routes is presented with the thickness of the trajectory, and on this basis, the source of the trajectory (identified by the dot) and the destination (identified by the triangle) are added and the different regions are represented with different colors. By these visual means, we can better analyze the clustering results.

After all the parameters are determined, the clustering analysis for 7-days GPS trajectory dataset is carried out, and the clustering results are shown in Table 3. The popular routes of the weekends are significantly more than those of weekdays, which is consistent with people's real travel laws. In weekends, the travel purposes of most of residents focus on shopping, leisure, entertainment and so on, so travel activities are randomness and clustering results are relatively dispersed. But in weekdays, most of residents mostly commute from the worksite to residence, so the clustering results are more concentrated.

According to the clustering results of GLTC algorithm, we analyze the flow direction of the popular routes synthetically. During morning peak hours of weekdays, the sources of trajectories are mainly distributed in Wuhua district, and the destinations are mainly distributed in the Guandu district, while at the afternoon peak hours, the conclusion is opposite. During weekends, the main sources and destinations of trajectories are distributed in Wuhua district, which indicates that this district provides most of places of leisure and entertainment for residents in Kunming.

The clustering results of trajectories from 21:00–22:00 in August 17, 2012 (weekdays) are shown in Fig. 5. From this figure we can discovery that the popular routes include Xinwen road, Dongfeng west road, Nanping Street and so on, consistent with the reality. This shows that GLTC algorithm not only retains the original spatio-temporal characteristics of the trajectory, but also describes the flow characteristics and behavior patterns of residents more fully.

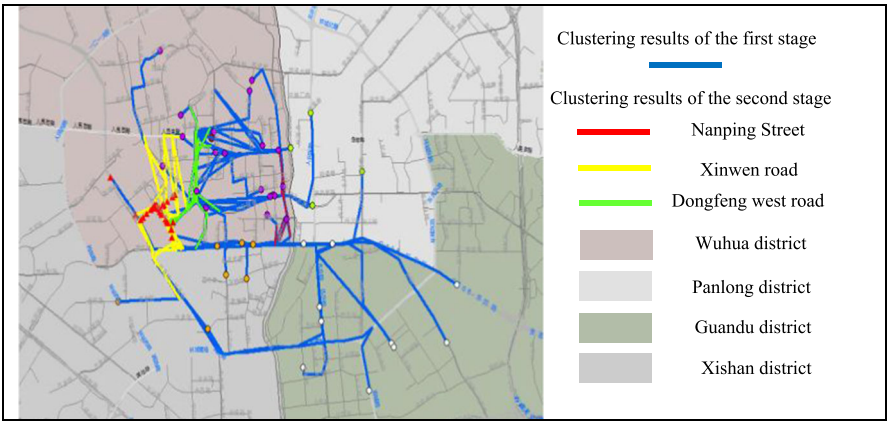


Fig. 5 Clustering results of *delta*=10, *sim*=0.95, $\omega = 10$ (August 17, 2012 21:00–22:00)

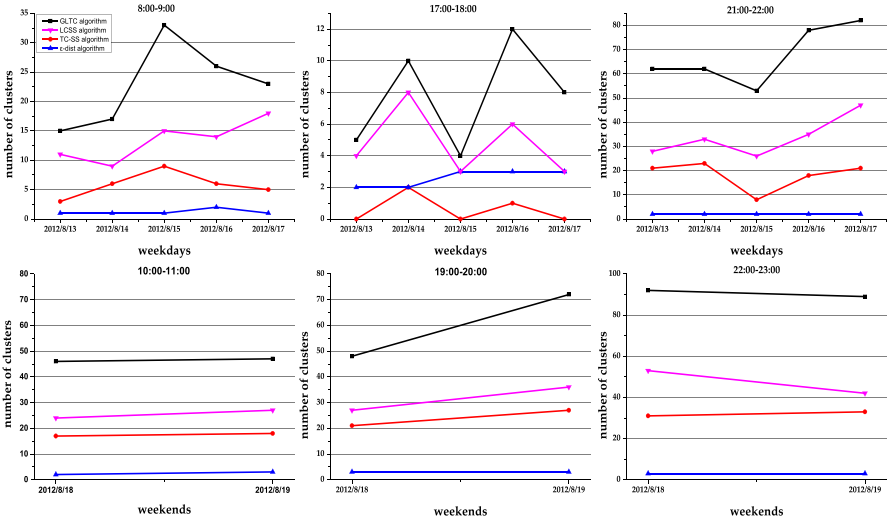


Fig. 6 Comparison of clustering results of four algorithms on different dates (weekdays and weekends)

4.4 Performance Comparison of Similar Algorithms

In this subsection, GLTC algorithm is compared with ε-dist algorithm in the literature [10], TC-SS algorithm in the literature [11] and LCSS algorithm in the literature [20], the results are shown in Fig. 6.

From Fig. 6, the number of clusters of TC-SS algorithm and ε-dist algorithm are few in different periods on weekdays; on weekends, the number of clusters of TC-SS algorithm has increased, but the difference is not large, and the number of clusters of ε-dist algorithm has no change. GLTC algorithm and LCSS algorithm, whether on weekdays or weekends, the number of clusters have similar trends, but the number of clusters have obvious gaps. The results show that the LCSS algorithm, TC-SS

Table 4 Davies-Bouldin index statistics table

Algorithm	Davies-Bouldin index
GLTC	0.57
LCSS	0.85
TC-SS	0.25
ε-dist	2.79

algorithm and the ε-dist algorithm only consider the partial structural features of the trajectory data and cannot acquire the exact location of popular routes accurately; but GLTC algorithm considers the structure features of trajectory data comprehensively, and can further discover the movement laws and patterns hidden in the trajectory.

It can be seen from the number of clusters that the LCSS algorithm, the TC-SS algorithm and the ε-dist algorithm have a large gap with the GLTC algorithm. For the unsupervised data, Davies-Bouldin (DB) index [21] is used to evaluate the effectiveness of the clustering results, which can be calculated with Eqs. 15. Where k represents the number of clusters, $\overline{C_i}$ and $\overline{C_j}$ represent the average trajectory distance (i $\neq$ j) interior clusters i and j, and w_i and w_j represent the central trajectories of the clusters i and j. The smaller the DB value means that the smaller the distance inside the cluster, the greater the distance between clusters.

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{\overline{C_i}+\overline{C_j}}{||w_i - w_j||_2}\right) \tag{15}$$

As can be seen from Table 4, the TC-SS algorithm obtains the smallest Davies-Bouldin index because there is no higher compactness within the cluster, and the reason why the ε-dist algorithm obtains the largest Davies-Bouldin index is that the distance between the clusters is small, and even some overlap between the different clusters, resulting in poor effectiveness of clustering results. The *DB* value of LCSS algorithm is slightly higher than the one of the GLTC algorithm, because the clustering number widens the distance between the clusters, while the GLTC algorithm mines a relatively large number of clustering results, at the same time obtaining a relatively small Davies-Bouldin index, which makes the clustering effect better. Finally, we can make the following conclusions: (1) The reliability and accuracy of clustering results using TC-SS algorithm, ε-dist algorithm and LCSS algorithm are lower than the ones of GLTC algorithm. (2) Because the GLTC algorithm needs to perform two stages clustering process, its execution time is relatively long compared with the other algorithms.

5 Conclusions

From the view of the global and local structural characteristics of trajectory data, we propose a GLTC trajectory clustering algorithm by combining the above two characteristics, adding direction, corner and position attributes, considering the full structural

characteristics of trajectories. Experimental results show that this algorithm can obtain more plentiful and accurate clustering quantity. In addition, this study displays clustering results using visualization methods. For example, different colors and shapes are used to display the different sources and destinations of trajectories. Trajectory thickness indicates different the number of trajectories. These methods can assist researchers and managers to better grasp the knowledge and laws in trajectory clustering results, and provide more valuable information to applications such as traffic management, route recommendation and city planning. The time complexity of this algorithm is $O(n^2)$, resulting in the computation of clustering results is relatively slow. Subsequently, we will reduce time complexity by adding space indexes and shorten the operation time of this algorithm at the cost of space.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant No. 61663047 and No. 61363084.

References

1. Vandenberghe W, Vanhauwaert E, Verbrugge S et al (2012) Feasibility of expanding traffic monitoring systems with floating car data technology. IET Intel Transp Syst 6:347–354
2. Tseng PJ, Hung CC, Chang TH, et al (2012) Real-time urban traffic sensing with GPS equipped probe vehicles. ITS Telecommunications (ITST), In IEEE 12th international conference on pp 306–310
3. Zheng Y (2015) Introduction to urban computing. Geomat Inf Sci Wuhan Univ 40:1–13
4. Davies JJ, Beresford AR, Hopper A (2006) Scalable, distributed, real-time map generation. IEEE Pervasive Comput 5:47–54
5. Camargo SJ, Robertson AW, Gaffney SJ, et al (2004) Cluster analysis of western North Pacific tropical cyclone tracks. In Proceedings of the 26th conference on hurricanes and tropical meteorology pp 250–251
6. Kharrat A, Popa IS, Zeitouni K, et al (2008) Clustering algorithm for network constraint trajectories. Headway in spatial data handling. In International symposium on spatial data handling, Montpellier, France, 23–25 July. DBLP pp 631–647
7. Roh GP, Hwang S (2010) Nncluster: An efficient clustering algorithm for road network trajectories. In International conference on database systems for advanced applications. Springer Berlin Heidelberg pp 47–61
8. Vlachos M, Kollios G, Gunopulos D (2002) Discovering similar multidimensional trajectories. Data Engineering, 2002. Proceedings. In 18th international conference on. IEEE pp 673–684
9. Pelekis N, Kopanakis I, Marketos G, et al (2007) Similarity search in trajectory databases. Temporal Representation and Reasoning. In 14th international symposium on. IEEE pp 129–140
10. Shen Y, Zhao L, Fan J (2015) Analysis and visualization for hot spot based route recommendation using short-dated taxi GPS traces. Information 6:134–151
11. Yuan G, Xia SX, Zhang L et al (2011) Trajectory clustering algorithm based on structural similarity. J Commun 9:103–110
12. Lee JG, Han J, Whang KY (2007) Trajectory clustering: a partition-and-group framework. Proceedings of the 2007 ACM. In SIGMOD international conference on Management of data. pp 593–604
13. Buchin M, Driemel A, van Kreveld M et al (2011) Segmenting trajectories: a framework and algorithms using spatiotemporal criteria. J Spat Inf Sci 3:33–63
14. Nawaz T, Cavallaro A, Rinner B (2014) Trajectory clustering for motion pattern extraction in aerial videos. In IEEE international conference on image processing (ICIP). IEEE pp 1016–1020
15. Chang C, Zhou B (2009) Multi-granularity visualization of trajectory clusters using sub-trajectory clustering. In IEEE international conference on data mining workshops. IEEE pp 577–582
16. Kumar KM, Reddy ARM (2016) A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. Pattern Recogn 58:39–48
17. Chen W, Zhang S, Lu AD (2013) The Fundamentals and Methods of Data Visualization. Science Press, Beijing

18. Ren L, Du Y, Ma S et al (2014) Visual analytics towards big data. J Softw 25:1909–1936
19. Pu SJ, Qu HM, Ni MX (2012) Survey on visualization of trajectory data. J Comput-Aided Des Comput Gr 24:1273–1282
20. Pei J, Peng DL (2016) LCSS-based computing similarity between trajectories for vehicles. J Chin Comput Syst 6:1197–1202
21. Bezdek JC, Pal NR (1998) Some new indexes of cluster validity. In IEEE transactions on systems, man, and cybernetics. Part B (Cybernetics) 3:301–315