Research on Trajectory Clustering Algorithm for Public Opinion Users

Fuyu Lu[1, a], Yonglin Leng[2,b], and Xiaohong Sun[3,c]

[1]Information Construction and Management Center, Bohai University, Jinzhou 121000, China

[2]College of Information Science and Technology, Bohai University, Jinzhou 121000, China

[3]College of Mangement, Bohai University, Jinzhou 121000, China

[a]lufuyu@qq.com; [b]lengyonglin@qq.com; [c]sxh@qq.com

Keywords: Trajectory clustering; AP algorithm; Public Opinion; Similarity

Abstract: Clustering movement trajectories to get the motion feature of object is one of the goals of the trajectory clustering. Aiming at the large scale trajectory data, to address the low efficiency of clustering, this paper proposes a hierarchical trajectory clustering algorithm based on time series (HTCTS). The algorithm first divides trajectory data by time interval, and then respectively cluster the sub paths. Finally, for all cluster subset, HTCTS executes cluster algorithm again to produce the final clustering results. The experimental results show that HTCTS algorithm in clustering efficiency and quality is superior than the trajectory clustering algorithms which cluster the whole dataset directly.

1. Introduction

Trajectory data is an important data in the monitor of public opinion. It comes from a wide range of sources, such as the activity route of criminal suspect, traffic route of large-scale gathering and so on. These trajectory data usually include time, speed and location of activities. Through the analysis of these trajectory data, we can get the movement patterns of individual or group, such as routine routes, road traffic and so no. All these can assist researchers and managers in making decision analysis [1, 2].

Trajectory clustering is an effective method for analyzing trajectory data. For different trajectory data, a large number of trajectory clustering algorithms have been studied. For example, Caffney et al. propose the model-based trajectory clustering algorithm [3-5]. Lee et al. propose TRACLUS, which is a density-based clustering algorithm implemented on all sub-trajectory sets [6]. Ahmed Kharrat et al. proposed NETSCAN, a density-based trajectory clustering method in road network space. NETSCAN first calculates busy paths according to the path traveled by mobile objects, and then clusters sub-trajectories according to the density parameters set [7]. Xia Ying et al. [8] propose a trajectory similarity measurement method considering time and space constraints under the constraints of road network.

Due to the increasing amount of trajectory data, the traditional clustering algorithm can not effectively cluster large-scale trajectory data. This paper proposes a hierarchical trajectory clustering algorithm based on time series (HTCTS). The algorithm first divides a complete trajectory data into different time intervals, and then the data in the same time interval are clustered to form clustering subset. Finally, the clustering subset is clustered to generate the corresponding clustering results.

2. Relevant Concepts

2.1 Trajectory Clustering.

Let $TR=\{TR_1, TR_2,...,TR_n\}$ is a set of trajectory paths, where each TR_i is a complete trajectory path. A trajectory path is usually a broken line, which consists of a series of continuous changing position sequences. Let $TR_i = P_i^1 P_i^2 \ldots P_i^m$, where P_i^j $(1 \leq j \leq m)$ represents a time feature point, and m represents the number of line segments in the trajectory path. In order to reduce the data size, this

  DOI: 10.25236/meici.2019.074

paper uses the trajectory path algorithm in reference [7] to reduce the number of midpoints in the path. That is, when the deviation value of a point is lower than a threshold value, the point can be neglected. The simplified trajectory path is represented by $TR_i^{simplified} = S_i^1 S_i^2 \dots S_i^q$, where S_i^j and S_i^{j+1} $(1 \leq j \leq q,\ q \ll m)$ connect end to end to form a sub-trajectory. It can also be expressed as a $TR_i^{simplified} = L_i^1 L_i^2 \dots L_i^{q-1}$, which L_i^j represents a sub-trajectory.

Given the set of trajectory paths *TR*, our goal is to cluster sub-trajectories and generate a cluster set, namely $C=\{C_1, C_2, \dots C_{cn}\}$, where $C_t = \{L_i^1, L_i^2, \dots, L_i^{ln}\}$. Each element in clustering comes from a sub-trajectory in a certain trajectory path. Through clustering sub-trajectory, we can find the common characteristics of object motion. It has great practical significance for solving practical problems, such as finding the best path, avoiding traffic congestion and analyzing user behavior.

2.2 AP Clustering Algorithm.

AP clustering is done by calculating the similarity between *N* data points. The similarity between *N* data points can be expressed by a *N*N* matrix *S*. Unlike other clustering algorithms, AP clustering does not need to specify the cluster centers. In the process of clustering, AP algorithm considers all data points as the potential cluster centers. The *k-th* data point can become a clustering center is determined by the corresponding diagonal line value $S(k, k)$. The greater the value of $S(k, k)$ is, the greater possibility of *k-th* point becoming a clustering center. This value is also called the reference degree. The number of clustering is affected by the reference degree. Initially, it is considered that each of them is affected by *p*. If *p* takes the mean of input similarity, then the number of clusters generated is medium. The smaller value of *p* is, the less number of clusters is generated [9].

AP clustering algorithm transmits two kinds of messages between data points through iteration, which are responsibility and availability. These two kinds of messages correspond to matrix *R* and *A*, respectively. $R(i, k)$ denotes the message sent from data *i* to candidate clustering center *k*, which reflects if the data *k* is suit for clustering center of data *i*, and $A(i, k)$ represents the message sent from candidate clustering center *k* to data point *i*, reflecting whether data *i* chooses *k* as its clustering center. The greater $R(i,k)$ and $A(i,k)$, the more likely data point *k* is to be the clustering center, and the easier point *i* chooses *k* as its clustering center.

AP clustering updates the responsibility and availability matrix by iteration until get *k* high quality exemplars. The responsibility matrix is updated by availability matrix and similarity matrix.

$$R(i,k)=S(i,k)-\max_{k' \neq k}\{A(i,k')+S(i,k')\} \tag{1}$$

The update of availability matrix is achieved by the responsibility matrix.

$$A(i,k)=\min\{0,R(k,k)+\sum_{i' \notin \{i,k\}}\max\{0,R(i',k)\}\}$$

$$A(k,k)=\sum_{i' \neq k}\max\{0,R(i',k)\} \tag{2}$$

3. A Hierarchical Trajectory Clustering Algorithm based on Time Series

In this paper, a trajectory clustering algorithm based on time series is proposed. Because the correlation degree between trajectory paths in different time interval is much less than that the trajectory paths in intervals, so the algorithm divides a complete trajectory data according to a certain time first. Then the data in each time interval is processed by AP and generates corresponding clustering subsets. In order to achieve the overall clustering of trajectory data, the algorithm carries out quadratic AP clustering on clustering subsets to get the final clustering results.

3.1 Similarity Measure.

The basis of clustering is the similarity between objects. There are two kinds of similarity measure methods in this paper. One is the similarity of sub-trajectories, the other is the similarity of sub-clusters.

Li et al. consider that the similarity between two sub-trajectories consists of three parts: central point similarity ($simi_{center}$), angle similarity ($simi_{\theta}$) and parallel similarity ($simi_{parallel}$), such as Eq.3,

where L_1 represents longer sub-trajectories and L_2 represents shorter sub-trajectories.

$$simi(L_1, L_2)=simi_{center}(L_1, L_2)+simi_{\theta}(L_1, L_2)+simi_{parallel}(L_1, L_2) \quad (3)$$

The similarity of the central point is calculated by Euclidean distance.

$$simi_{center}(L_1,L_2)=\|L_1^{center} - L_2^{center}\| \quad (4)$$

In the calculation of angle similarity, the smaller intersection angle between two sub-trajectories is expressed by θ. Since we do not consider the direction of sub-trajectories in this paper, so the value of θ satisfies $0°\leq\theta\leq180°$. Eq.5 gives the calculation of angle similarity of sub-trajectories.

$$simi_{\theta}(L_1, L_2) = \begin{cases} \|L_1\| \times \sin(\theta) & 0° \leq \theta \leq 90° \\ \|L_2\| & 90° \leq \theta \leq 180° \end{cases} \quad (5)$$

Let L_1^s and L_1^e be the starting and ending points of sub-trajectory L_1, and L_2^s and L_2^e correspond to the two endpoints of sub-trajectory L_2, respectively. In addition, we use p_s and p_e to represent the projection of two endpoints of L_2 on L_1.

$$para_1 = \|p_s - L_1^s\| \quad (6)$$

$$para_2 = \|p_e - L_1^e\| \quad (7)$$

The parallel similarity of the two sub-trajectories is the minimum value of $para_1$ and $para_2$, as shown in Eq. (8).

$$simi_{parallel}(L_1, L_2)=\min(para_1, para_2) \quad (8)$$

The triple $TL=\{N, LS_{\text{center}}, LS_{\theta}, LS_{length}\}$ describes the cluster subset information of sub-trajectory clustering, where N represents the number of sub-trajectory, LS_{center}, LS_{θ} and LS_{length} represent the sum of all sub-trajectory centers, angles and lengths, respectively.

Because the sub-trajectories similarity in sub-trajectories clustering set are very large. So when calculating the similarity between sub-trajectories clustering sets, we select a representative from the sub-trajectories clustering set, and then calculate the similarity among the representatives as the similarity of sub-trajectories clustering sets. In Eq.9 and Eq.10, R_s and R_e represent the starting and ending points of sub-trajectories.

$$R_s = (center_x - \frac{cos\theta}{2} len, center_y - \frac{sin\theta}{2} len) \quad (9)$$

$$R_e = (center_x + \frac{cos\theta}{2} len, center_y + \frac{sin\theta}{2} len) \quad (10)$$

where $center_x=LS_{center}^x/N$, $center_y = LS_{center}^y/N$, $len = LS_{length}/N$, $\theta = LS_{\theta}/N$.

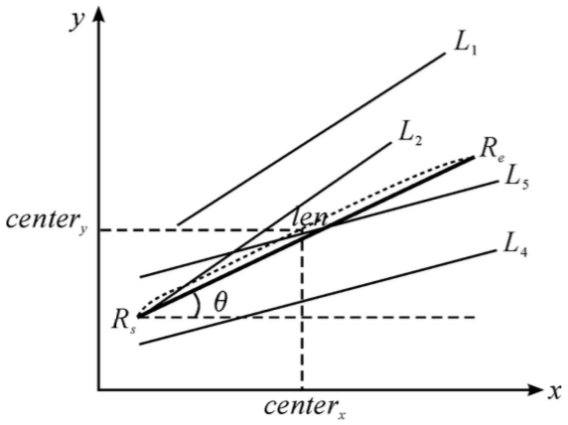

Figure 1. Representative sub-trajectory

3.2 HTCTS Algorithm.

Given the set of trajectory paths TR and the time interval parameter t, the steps of HTCTS algorithm are as follows:

Step 1: according to the given time interval, divide the trajectory path set TR, and simplify the trajectory path to a sub-trajectory set;

Step 2: according to Eq.3, calculate the similarity of sub-trajectories in each sub-trajectory set, cluster subsets by AP clustering;

Step 3: find representative sub-trajectories of each cluster subset;

Step 4: compute the similarity between representative sub-trajectories;

Step 5: cluster representation sub-trajectories using AP clustering algorithm;

Step 6: distribute the corresponding to the cluster where the sub-trajectories belong;

Step 7: Output the clustering result C.

4. Experiment

In order to test the performance of HTCTS algorithm, the experimental environment chooses Windows XP professional operating system. The hardware configuration includes Intel Xeon 2.00GHz, 4GB memory and 500G hard disk. The HTCTS algorithm is written by C++ and compiled by gcc. The experimental dataset is a Geolife project led by the Academy of Sciences including 182 users' trajectory data from April 2007 to August 2012. This paper randomly selects 100 users' trajectory data from the dataset for experiments in one day.

4.1 Evaluation of Cluster Quality.

The experiment divides the trajectory data by a certain time interval, and then clusters the trajectory data by HTCTS algorithm. Table 1 lists the trajectory path clustering time and the number of clusters at different time intervals. Experiment results show the clustering time interval has little effect on the clustering efficiency. For example, the trajectory path is clustered in an hour time interval. The clustering time is 2.67s, when the time interval is reduced to half an hour, and the clustering time only increases by 0.16s. The reason is that the number of sub-trajectories in a short time interval is lower than that in a long time interval, and the amount of similarity calculation and the size of cluster sub-trajectories decrease for each cluster sub-trajectory. Although the number of clusters increase, but the number of iterations is lower than that of long interval clustering. Because the amount of data in each cluster is much smaller than that in long interval clustering, so the time change is not obvious.

In this experiment, the number of clustering only includes that the number of sub-trajectories is higher than a certain threshold. The other clustering results which are lower than the threshold are regarded as noise areas, that is, these clustering results do not share the features all users. The result shows that the setting of time interval will affect the change of clustering number. Because AP clustering does not need to set the number of clustering. When the time interval is long, some sparse trajectories are classified as noise areas, so the number of clusters is lower than that of clusters with short time intervals.

Table 1 Clustering quality of HTCTS

Time interval [h]	Clustering Time [s]	Clustering Number
1	2.67	7
0.5	2.83	8
0.25	3.05	10

4.2 Comparison of Similar Algorithms.

Fig. 2 compares HTCTS with TRACLUS and DBSCAN clustering algorithms. It can be seen that HTCTS has better efficiency in clustering trajectory paths, and the number of clustering is more than other algorithms. The main reason is that HTCTS divides trajectory paths into subsets by time. The correlation of trajectory paths at different time intervals is not significant, so this division does not affect the clustering quality, but the time efficiency has been improved.

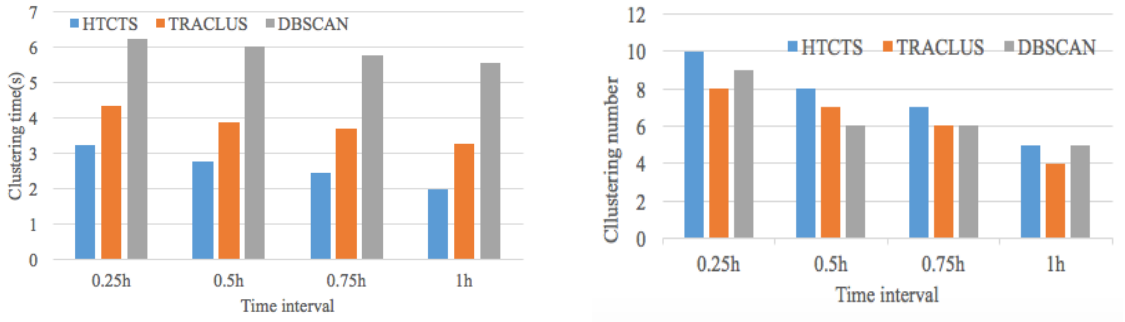

Figure 2. Comparision of different algorithms

5. Conclusion

Traditional trajectory clustering algorithms mostly focus on the complete trajectory path. Because the large amount of trajectory data would greatly reduce the efficiency of clustering, and greater correlation of trajectory paths within a certain time interval. This paper proposes a hierarchical trajectory clustering algorithm based on time series. HTCTS divides trajectory paths into two parts. The first is to cluster the sub-trajectories within each time interval. The second is to cluster the results of the sub-trajectory clustering as a whole. Experiments verify the influence of different time intervals on clustering results, and compare HTCTS algorithm with the whole trajectory clustering algorithm. The results show that the clustering efficiency and clustering quality of HTCTS are greatly improved.

Acknowledgements

This work is partly supported by Liaoning Social Science Fund (No. L14AGL002, L13AGL002) and the Science and Technology Project of the Liaoning Provincial Education Department (No. LQ2017004).

References

[1] S. Wang, R. Sinnott and S. Nepal: *Proc. IEEE International Conference on Big Data* (Boston, MA, USA, December 11-14, 2017). Vol.1, p.1109.

[2] G. Yuan, S.X. Xia, L. Zhang: *Journal on Communications*, Vol. 32 (2011) No.9, p.103.

[3] S. Gaffney and P. Smyth: *Proc. the 5th International Conference on Knowledge Discovery and Data Mining* (San Diego, CA, USA, August 15-18, 1999). Vol.1, p. 63.

[4] I.V. Cadez, S. Gaffney and P.A. Smyth: *Proc. the 6th International Conference on Knowledge Discovery and Data Mining* (Boston, MA, USA, August 20-23, 2000). Vol.1, p.140.

[5] S. Tasnim, J. Caldas and N. Pissinou. *Proc. International Conference on Computing, Networking and Communications* (Maui, HI, USA, March 5-8, 2018). Vol.1, p.88.

[6] J.G. Lee, J. Han and K.Y. Whang: *Proc. the ACM SIGMOD International Conference on Management of Data* (Beijing, China, June 12-14, 2007). Vol.1, p. 593.

[7] A. Kharrat, I.S. Popa, K. Zeitouni: *Proc. International Symposium on Spatial Data Handling* (Montpellier, France, July 23-25, 2008). Vol.1, p. 631-647.

[8] Y. Xia, G.Y. Wang AND X. Zhang: *International Journal of Computational Intelligence Systems*, Vol. 4 (2010) No. 5, p. 491.

[9] B.J. Frey, D. Dueck: *Science*, Vol. 315 (2007) No. 5814, p. 972.