Clustering of Trajectories Based on Hausdorff Distance

Jinyang Chen [1,2] Rangding Wang [1+]
[1]Ningbo University
College of Information Science and Engineering
Ningbo China
E-mail:jeekboy1@163.com

Liangxu Liu [2] Jiatao Song [2]
[2] Ningbo University of Technology
School of Electronic and information
Ningbo China
E-mail:luransh@gmail.com

+ Corresponding author: E-mail: wangrangding@nbu.edu.cn

Abstract —**Spatio-temporal and geo-referenced datasets are growing rapidly, with the rapid development of some technology, such as GPS, satellite systems. At present, many scholars are very interested in the clustering of the trajectory. Existing trajectory clustering algorithms group similar trajectories as a whole and can't distinguish the direction of trajectory. Our key finding is that clustering trajectories as a whole could miss common *sub*-trajectories and trajectory has direction information. In many applications, discovering common sub-trajectories is very useful. In this paper, we present a trajectory clustering algorithm CTHD (clustering of trajectory based on hausdorff distance). In the CTHD, the trajectory is firstly described by a sequence of flow vectors and partitioned into a set of sub-trajectory. Next the similarity between trajectories is measured by their respective Hausdorff distances. Finally, the trajectories are clustered by the DBSCAN clustering algorithm. The proposed algorithm is different from other schemes using Hausdorff distance that the flow vectors include the position and direction. So it can distinguish the trajectories in different directions. The experimental result shows the phenomenon.**

Keywords-trajectories cluster; flow vector; hausdorff distance

I. INTRODUCTION

Recent improvements in satellites and tracking facilities have made it possible to collect a large amount of *trajectory* data. There is increasing interest to perform data analysis over these trajectory data. A number of clustering algorithms have been reported in the literature, the traditional clustering algorithms are mainly divided into base on partition method (k-means [1]), based on hierarchy method (BIRCH [2]), based on density method (DBSCAN [3]), based on grid method (CLIQUE [4]) and Based on model method (COBWEB [5]).

According to different target object, clustering analysis in spatio-temporal dataset can be divided into based on mobile object [6-10] and based on the trajectory [11-15]. Trajectory clustering algorithms group similar trajectories as a whole, thus discovering common trajectories. D. Chudova et al. [11] had proposed a model-based clustering algorithm for trajectories through space-time transformation, in which trajectory was expressed as a curve model by the mixture model and EM algorithm was proposed in this foundation. However, the points of trajectory not only are growing rapidly, but also will be more in the near future. So these methods can't satisfy all trajectory clustering requirements. Clustering trajectories as a whole possibly led to miss common *sub*-trajectories, so discovering common sub-trajectories is very useful in many applications,such as weather forecast and traffic control. In order to solve this problem, Lee [11] had proposed a *partition-and-group* framework in which trajectories are divided into sub-trajectories as clustering of object, sub- trajectories were clustered by using density of clustering methods.

Our key observation is that trajectory not only includes the position information but also the moving direction. If the trajectories are clustered only using the position, trajectories which had similar shape and different directions, couldn't be distinguished. As figure 1, entire space consists two sub-spaces R_1 and R_2, S_4,S_5,S_6 are three

978-1-4577-0321-8/11/$26.00 ©2011 IEEE

trajectories in the sub-space R_2, They belong to a class because of their shape and move direction is similar. S_1,S_2,S_3 are three trajectories in the sub-space R_1, the move direction of S_2 is different of S_2 and S_3. If only use position information to cluster those trajectories, S_2 will similar to S_1 and S_3. So we will get worry result.

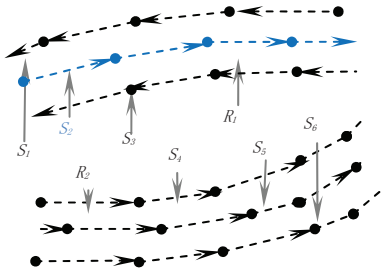

Figure 1. Trajectory characteristics

In this paper, we take moving direction of trajectory into account and propose a trajectory clustering algorithm CTHD (clustering of trajectory based on hausdorff distance). In which trajectory is looked as the time sequence points, so the trajectory could be described by a sequence of flow vectors and partitioned into a set of sub-trajectory. Next the similarity between sub-sequences is measured by their respective Hausdorff distances. Finally, the trajectories are clustered by the DBSCAN [3].

The rest of the paper is organized as follows. Section 2 presents a trajectory description. Section 3 presents an overview of our trajectory clustering algorithm. Section 4 presents the results of experimental evaluation. Section 5 concludes the paper.

II. TRAJECTORY DESCRIPTION

Given a set of *trajectories* $I = \{ TR1; \ldots ; TR_{\mathrm{num_{tra}}} \}$, num_{tra} is number of trajectory. A *trajectory* is a sequence of 2-dimensional points. It is denoted as $TR_i = p_1p_2p_3\ldots p_j\ldots\ p_{len_i}$ ($1<i<num_{tra}$). Here, p_j ($1<j<len_i$) is a 2-dimensional point. One trajectory TR_i in the 2D, can be given as:

$$TR_i = \{(x_1, y_1), (x_2, y_2)\ldots (x_n, y_n)\} \qquad (1)$$

Instead of using a sequence of positions to describe an object's movements, we describe its trajectory in terms of a sequence of *flow vectors* where a flow vector f represents:

$$f_i = (x_i, y_i, dx_i, dy_i)$$

where

$$\mathrm{d}x_i = (x_{i+1} - x_i) / \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \qquad (2)$$

$$\mathrm{d}y_i = (y_{i+1} - y_i) / \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \qquad (3)$$

The direction components are scaled relative to the positional components in order to balance their relative contribution when computing the similarity between flow vectors. Flow vectors are transformed so that each component lies in the range [0, 1]. The trajectory is described by a sequence of flow vectors F:

$$F = \{ f_1 , f_2 , \cdots, f_l \} \qquad (4)$$

HD (Hausdorff Distance) is a measurement distance method between two points set (not matching degree). Different to other measurement method, HD is not established corresponding points. So it would neglect the difference between motion directions. So in this paper, the trajectory is described by a sequence of flow vector to avoid this problem. Therefore, hausdorff distance between two trajectories can be transformed into following type:

$$\begin{cases} H(F_i,F_j) = \max(h(F_i,F_j),h(F_j,F_i)) \\ h(F_i,F_j) = \max\limits_{f_a\in F_i}(\min\limits_{f_b\in F_j}(dist(f_a,f_b))) \\ h(F_j,F_i) = \max\limits_{f_b\in F_j}(\min\limits_{f_a\in F_i}(dist(f_b,f_a))) \end{cases} \qquad (5)$$

dist (f_a,f_b) is Euclidean distance between f_a and f_b, f_a, f_b is a flow vector.

III. CLUSTERING OF TRAJECTORIES OF MOVING OBJECTS

In this section, a sub-trajectory clustering algorithm is proposed for the grouping phase. Firstly the trajectory segmentation is discussed in Section A. Secondly we present density-based clustering algorithm for sub-trajectory, which method is based on the algorithm DBSCAN [3].

A. Trajectory Segmentation

Before we discuss trajectory segmentation the concept of *characteristic flow vector* is propose firstly.

Definition 1. If $f_i(x_i, y_i, dx_i, dy_i)$ satisfies $\theta=\sqrt{(dx_i-dx_{i-1})^2+(dy_i-dy_{i-1})^2}>\delta$, f_i is a *characteristic flow vector* , δ is a threshold value. The first and last *flow vector* of a sub-trajectory is *characteristic flow vector.*

From a trajectory $TR_i = \{f_1f_2f_3\ldots f_j \ldots f_{leni}\}$, we

determine a set of *characteristic flow vectors* $\{f_{c1}, f_{c2}, f_{c3}, \ldots, f_{cpari}\}$ $(c_1<c_2<\ldots<c_{pari})$, c_{prai} *is number of characteristic flow vector*. Then, the trajectory TR_i is partitioned at every *characteristic flow vector*. Figure 2 shows TR_i is partitioned into a set of 3 sub-trajectory $\{(f_1f_2f_3), (f_3f_4f_5), (f_5f_6f_7f_8)\}$.

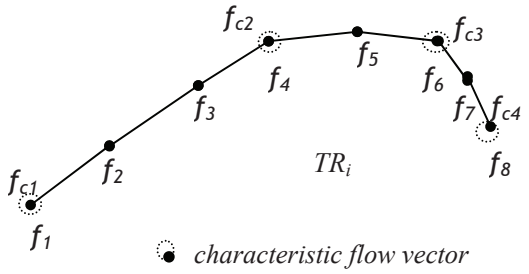

Figure 2: An example of a trajectory and its trajectory partitions.

B. Trajectory Clustering Based on DBSCAN

Originally, DBSCAN [3] designed to clustering for points, which can't be used in trajectory clustering directly. In this paper, some definitions in the DBSCAN are changed by sub-trajectory instead of point [11].

Definition 2. Sub-trajectory $L_j \in D$ is the ε -*neighborhood* of a sub-trajectory $L_i \in D$ if dist(L_i,L_j)$\leq \varepsilon$, record as $N_\varepsilon(L_i) = \{L_j \in D \mid dist(L_i, L_j) \leq \varepsilon\}$. D is the set of all sub-trajectories.

Definition 3. A sub-trajectory $L_i \in D$ is called a *core sub-trajectory* w.r.t. ε and *MinLns* ,if the number of ε *-neighborhood* of a sub-trajectory L_i more than threshold value *MinLns,* record as $|N_\varepsilon(L_i)| \geq MinLns$.

Definition 4. A sub-trajectory $L_i \in D$ is *directly density-reachable* from a sub-trajectory $L_j \in D$ w.r.t. ε and *MinLns.* if (1) $L_i \in N_\varepsilon(L_j)$ and (2) $|N_\varepsilon(L_j)| \geq MinLns$

Definition 5. A sub-trajectory $L_i \in D$ is *density-reachable* from a sub-trajectory $L_i \in D$ w.r.t. ε and *MinLns* if there is a chain of sub-trajectories $L_j, L_{j-1}, \cdots, L_{i+1}, L_i \in D$,such that L_k is directly density-reachable from L_{k+1} w.r.t. ε and *MinLns*.

Definition 6. A sub-trajectory $L_i \in D$ is *density-connected* to a sub-trajectory $L_j \in D$ w.r.t. ε and *MinLns* if there is a sub-trajectory $L_K \in D$ such that both L_i and L_j are density-reachable from L_k w.r.t. ε and *MinLns*.

Definition 7. A non-empty subset $C \subseteq D$ is called a *density-connected set* w.r.t. ε and *MinLns* if C satisfies the following two conditions:

(1) *Connectivity*: $\forall L_i, L_j \in C$,L_i is density-connected to L_j w.r.t. ε and *MinLns*.

(2) *Maximality*: $\forall L_i, L_j \in D$, if $L_i \in C$ and L_j is density-reachable from L_i w.r.t. ε and *MinLns*, then $L_j \in C$.

In this formwork, we replace whole trajectory with sub-trajectory to act as clustering objects. And then the idea of DBSCAN is introduced to cluster sub-trajectories, DBSCAN which would find those clusters irregular shape and number of uncertainty, not sensitive to noise.

```
Input:  (1) A set of sub-trajectories D = {L1...Lnumln}
        (2) Two parameters ε and MinLns
Output: A set of clusters O ={C1...Cnumclus}
\*Step1*\
01: Set clusterId to be 0; /* an initial id */
02: Mark all the sub-trajectories in SetOfSegments as unclassifed;
03: CclusterId:=nextId(NOISE);
04: for each sub-trajectory Li ∈ D DO
05:   if Segment.ClId=UNCLASSIFIED then
06:   if ExpandCluster(SetOfSegments,Segment,
        clusterId,Eps,MinPts) then
07:       CclusterId:=nextId(Ccluster.Id)
\*Step2*\
ExpandCluster(SetOfSegments,Segment,CclusterID,Eps,MinPts)
08: seeds=SetOfSegments.regionQuery(Segment,Eps);
09: WHILE (seeds!=Empty) DO
10:   currentS=seeds.first();
11:   result=SetOfSegments.regionQuery(currentS,Eps);
12:   if result.size<MinPts then
13:    SetOfSegments.changeClId(resultS,NOISE);
14:   if result.size>=MinPts then
15:    for each sub-trajectory Lj ∈ result DO
16:      if resultS.ClId=UNCLASSIFIED then
17:        seeds.append(resultS);
18:        SetOfSegments.changeClId(resultS,CclusterID);
19:     seeds.delete(currentS);
```

Figure 3: A density-based clustering algorithm for sub-trajetories.

A density-based clustering algorithm for sub-trajectories is shown as Figure 3. Each trajectory was partitioned and put sub-trajectories into a set of sub-trajectories D as input. The algorithm CTHD was described as follows for each sub-trajectory.

Firstly, put sub-trajectory set into *SetOfSegments.* each pending sub-trajectory is initialized. Set *cluster Id* to be 0.

Secondly, a sub-trajectory L_i, which is arbitrarily selected, is removed from *SetOfSegments* and is judged whether it has already been processed, if it is false, *ExpandCluster*() is implemented. If *ExpandCluster*() return true, increase a new cluster.

Thirdly, the number of neighborhood of $L_i \in D$ is judged whether it is more than a given threshold, to determine whether it is selected to act as a core sub-trajectory. If L_i is not a core sub-trajectory, L_i is temporarily marked as a noise.

Lastly, if L_i is a core sub-trajectory, clustering object is density connected to L_i. the number of neighborhood of $L \in result$ is judged whether it is more than a given threshold, to determine whether it is selected to act as a core sub-trajectory. If L_j is a core sub-trajectory, L_j is allocated into cluster $C_{clusterId}$.

The time complexity of the algorithm in Figure 3 is $O(n \log n)$ if a spatial index is used, where n is the total number of sub-trajectories in a database. If we do not use any index, we have to scan all the sub-trajectories in a database. Thus, the time complexity would be $O(n2)$.

IV. EXPERIMEN TAL EVALUATION

A. *Experimental Environment and Data*

Experimental environment is Windows Server 2003 SP3 operating system, Intel Pentium Dual 2.10G CPU and development environment tool is Microsoft Visual Studio 6.0.

Experimental dataset is from real data, which includes information of hurricane longitude and latitude location, Maximum center of wind, center pressure, etc. We use the Atlantic hurricanes from 1850 to 2010, which includes 1294 trajectories that formed by 30368 points.

B. *Experimental Results and Analysis*

According to our heuristic, we set ε be 0.2 and *MinLns* be 5 to 7. Using visual inspection and domain knowledge, we are able to obtain the optimal parameter values; when $\varepsilon = 0.2$ and $MinLns = 6$.

In order to analyze the experiment results, we use *representative trajectories* to express the clusters [11]. Figure 4 shows the clustering result using the optimal parameter values. Thin green lines are trajectories, and thick red lines are *representative trajectories*. These representative trajectories are exactly *common sub-trajectories*. Here, from the number of red lines. We can obtain that five clusters are identified.

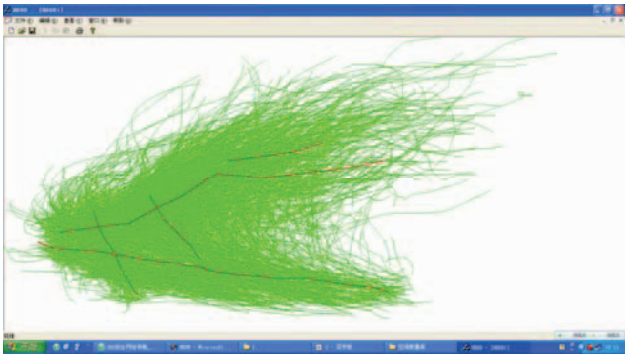

Figure 4. Clustering result for the hurricane data.

V. CONCLUSIONS

In this paper, we present a trajectory clustering algorithm, in which the trajectory is firstly described by a sequence of flow vectors and then partitioned into a set of sub-trajectory. Next the sub-trajectories are clustered by the DBSCAN. This algorithm solves the problem of trajectory direction. It takes trajectory direction into account when comparing the distance between two sub-trajectories. So it could find the difference between two sub-trajectories.

Acknowledgments

The work in this paper was partially supported by a project from the National Natural Science Foundation of china (Grant No.60972163, 60873220), five projects from Zhejiang Provincial Natural Science Foundation of China (No.Y1100589,Y1080123,Y108022,Z1090622,Y109028 5) , Zhejiang Science & Technology Preferred Projects of China (2010C11025), Zhejiang Province Education Department Key Project of China (ZD2009012), Ningbo Science & Technology Preferred Projects of China (2009B10003), Ningbo Key Service Professional Education Project of China (2010A610115), three projects from the Natural Science Foundation of Ningbo (Grant No. 2009A610090, 2010A610106, 2009A610085). and Ningbo University Foundation (XYL10002, XK1087). In addition, our programs are supported by High School Special Fields Construction of Computer Science and Technology (TS10860), Zhejiang Province

Excellent Course Project (2007), Zhejiang Province Key Teaching Material Construction Project (ZJB2009074).

VII.REFERENCES

[1] Marques JP, Written; Wu YF, Trans. "Pattern Recognition Concepts, Methods and Applications". 2nd ed., Beijing: Tsinghua University Press, 2002. 51−74 (in Chinese).

[2] ZHANG,T. RAMAKRISHNAN,R. and LIVNY, M. 1996." BIRSH: an efficient data clustering method for very large databases[C]". In Proceedings of the ACM SIGMOD Conference, 103-114,Montreal, Canada.

[3] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," In Proc. 2nd Int'l Conf. on Knowledge Discovery and Data Mining, Portland, Oregon, pp. 226-231, Aug. 1996.

[4] Zhao YC,Song J. "GDILC: A grid-based density isoline clustering algorithm" . In: Zhong YX,Cui S, Yang Y eds·Proc· of the Intemet Conf. onInfo-Net. Beijing: IEEEPress.2001:140-145P

[5] FISHER, D. 1987. "Knowledge acquisition via incremental conceptual clutering[J]". Machine Learning ,2, 139- 172.

[6] P. Kalnis, N. Mamoulis, S. Bakiras. "On Discovering Moving Clusters in Spatio-temporal Data". In Proceedings of the 9th International Symposium on Spatial and Temporal Databases, Angra dos Reis, Brazil, 2005, pp.364-381.

[7] S.-Y. Hwang, Y.-H. Liu, J.-K. Chiu, and E.-P. Lim. "Mining mobile group patterns: A trajectory-based approach". In Proceedings of the 9th Pacific–Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05), pp. 713–718. Springer, Berlin Heidelberg New York, 2005.

[8] M. Nanni and D. Pedreschi. "Time-focused density-based clustering of trajectories of moving objects". Journal of Intelligent Information Systems, 27(3): 267–289, 2006.

[9] Hwang J R, Kang H Y, Li K J "Spatio-temporal Similarity analysis between trajectories on road networks[C]". ER,2005： 280-289.

[10] Christian S. Jensen, Dan Lin, Beng Chin Ooi "Continuous Clustering of Moving Objects" Page(s): 1161-1174 Digital Object Identifier 10.1109/ TKDE. 2007.1054

[11] J. Lee, J. Han, and Kyu-Young Whang. "Trajectory clustering: A partition-and-group framework". In Proc. 2007 ACM SIGMOD Int'l Conf. on Management of Data, pages 593-604, Beijing, China, June 2007.

[12] Yingyi Bu , Lei Chen , Ada Wai-Chee , Fu Dawei Liu "Efficient Anomaly Monitoring Over Moving Object Trajectory Streams".*KDD'09,* June 28–July 1, 2009, Paris, France. Copyright 2009 ACM 978-1-60558-495-9/09/06

[13] D. Chudova, S. Gaffney, E.Mjolsness, and P. Smyth. "Translation-invariant mixture models for curve clustering". In Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining (KDD'03), pp. 79–88. ACM, New York, 2003.

[14] Zhenhui Li, Jae-Gil Lee, Xiaolei Li and Jiawei Han, "Incremental Clustering for Trajectories". Lecture Notes in Computer Science, 2010, Volume 5982/2010, 32-46, DOI: 10.1007/978-3-642-12098-5-3

[15] Zhen hui, Li Ming, Ji Jae-Gil Lee , Lu-An Tang , Yintao Yu , Jiawei Han , Roland Kays, " MoveMine: Mining Moving Object".Databases SIGMOD'10, June 6–11, 2010, Indianapolis, Indiana, USA. Copyright 2010 ACM 978-1-4503-0032-2/1