International Journal of Grid Distribution Computing
Vol.8, No.2 (2015), pp.1-8
http://dx.doi.org/10.14257/ijgdc.2015.8.2.01

A Trajectory Data Clustering Method Based On Dynamic Grid Density

Junhuai Li[1,2], Mengmeng Yang[1], Na Liu[1], Zhixiao Wang[1,2] and Lei Yu[1,2]
[1]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[2]Shaanxi Key Laboratory for Network Computing and Security Technology Xi'an 710048, China
junhuai.li@gmail.com

Abstract

Under the traditional method of frequent trajectory mining, the location of data is obtained through the GPS device. However, limited equipment accuracy may incur location ambiguity. In this paper, we propose a new trajectory data clustering method based on dynamic grid density, in order to remove this ambiguity. In this method, the trajectory space of an object is firstly divided into equal-sized squares dynamically. Then the trajectories of object are mapped to their corresponding square. Next, the density of each grid is calculated and all the frequent squares are acquired given the minimum support threshold. Lastly, the frequent area is obtained by merging the frequent squares acquired previously, using the boundary function provided. The experimental results show that this method provides an optional way of finding the frequent movement sequence.

Keywords: *Frequent trajectory, Clustering, Support, Density*

1. Introduction

With the rapid development of wireless communication and mobile computing technology, Location Based Service (LBS) has become one of the important requirements in our daily life. However, in practice, location ambiguity is incurred due to the accuracy limitation in positioning equipment; namely, the coordinate values of the same location obtained by the same positioning equipment at different time may differ from one another. In practical applications, since a trajectory is approximated using a series of discrete points, the more discrete points a trajectory contains, the more accurate the trajectory obtained will be. However, obtaining the position points of moving objects by high sampling rate will result in a large number of data acquired, which brings greater difficulty to data storage and analysis. This issue has been widely concerned with the scholars, namely, research in clustering method of trajectory data.

Trajectory data clustering refers to the aggregation of similar points to one class. There exists three main trajectory data clustering algorithms: the one based on DBSCAN (Density-based Spatial Clustering of Application with Noise) algorithm [2], the one on Simplified Character algorithm [3] and the one on Grid Partitioning algorithm [4]. Nikos Mamoulis, *et al.*, [2] apply the classic DBSCAN algorithm in the trajectory data clustering and calculate support value of each area after the completion of clustering, then find all frequent area according to the given value of minimum support threshold. In the methods of trajectories data clustering based on the simplified trajectory, the trajectories are usually simplified by the improved DP (Douglas Peucker) algorithm. This DP algorithm effectively reduces the number of trajectory data points by expressing the curve as a series of points. Moreover, the DP algorithm provides optimal approximate results [5-6] to the curve. In addition, Jae-Gil Lee, *et al.*, [6] also propose a method named the

Minimum Description Length (Minimum Description Length, MDL) principle to simplify the trajectories. They compare each segment after the object trajectories all are simplified into multiple segments. After that, they cluster the similar segments to an area, and use the region ID to represent all segments inside the region. Juyoung Kang, *et al.*, [5] adopt an improved DP algorithm named DPHull to simplify the trajectories, this method converts trajectory fragments to eigenvector for processing, and uses BIRCH algorithm to cluster the trajectory segments after the completion of the above conversion. In addition, in order to incorporate the time factors existing in the trajectory, Juyoung Kang [5, 8] transformed the trajectory fragments into eigenvector with time characteristic, and standardization the eigenvector by minimum– maximum principle. Lastly, they clustered the eigenvectors using BIRCH algorithm. Besides, we could also use MDL standard to simplify the trajectories [6], and DBSCAN clustering algorithm based on density clustering to trajectory segments.

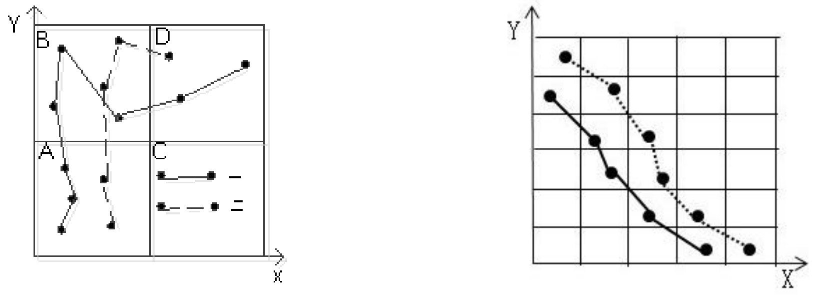
Although DBSCAN algorithm shows good performance in spatial data clustering, it causes huge memory consumption and I/O overhead, with a time complexity of *O*(N2) in the worst case scenario. Therefore, within the range of allowable error, a trajectory data clustering algorithm based on grid partitioning has been presented in literature [4, 9-11], in order to reduce the time consumption and improve the clustering speed. Ilias Tsoukatos, *et al.*, [4] discuss the spatial and temporal pattern problem existed in data mining environment under different granularity space. They obtain higher level of granularity space by merging sub-squares of low-level. Though this method implements space division of different granularity by combing the sub-grids, it only merges the adjacent four small squares to form larger space areas.

Fosca Giannotti, *et al.*, [10] propose a more adaptive method, which maps the trajectories to the squares and calculates the density of each square. If the density of a square is greater than the given minimum support threshold, then the squares is considered a frequent squares. The frequent area could be then generated by merging frequent squares obtained. N.S Savage, *et al.*, [9] also use the trajectory data clustering algorithm based on grid partitioning for frequent paths mining of GPS data. Differing the two methods discussed previously, they avoid converting the object trajectories into a sequence of square ID. Rather, they convert the trajectories into a sequence of grid side IDs. If a path through a side of a square, then this edge density increases by one. The frequent trajectories of users are formed by selecting *k* edges of the maximum density value.

2. Clustering Method based on Dynamic Grid Division

2.1. Problem Analysis

The performance of the trajectory data clustering method based on grid partitioning discussed previously is susceptible to the grid size selection. If the grid size is too large, the trajectories will be lost. As is shown in Figure 1(a), the trajectories in area B are lost, and thus both trajectories are converted to (A->B->D) at last. On the other hand, if the grid size chosen is too small, two similar points may fall into two different grids. Therefore, two similar trajectories could be converted into completely inconsistent, as is shown in Figure 1(b).



(a) Coarse-grained Squares **(b) Fine-grained Squares**

Figure 1. Trajectory Simplification with Fixed Size Cells

2.2. Trajectories Data Clustering Method

In order to solve the problem mentioned above, this paper proposes a trajectory data clustering method based on dynamic grid division-merger limit. Firstly, we divide the object trajectory space into equal-sized squares and map the object trajectories into these squares, with each points represented by the square ID. Then, we calculate the density of each grid and find all frequent squares according to the given minimum support threshold. Lastly, the frequent area is obtained by merging frequent squares based on the boundary function provided.

In this method, two parameters are introduced: square side length *d*, the value of which could be defined dynamically; and boundary function B(*m*,*n*), which is used as an upper-limit to the size of an area formed by square merging. The details are as follows:

(1) The trajectory space of all object is divided into many grids size for $d\times d$, where the value of *d* is defined dynamically. The optimal value of *d* could be determined before the start of the clustering; rather, it is determined through the sampling analysis of experimental data in experimental analysis phase. The optimal value of *d* corresponding to different data sets may vary.

(2) The object trajectories are mapped to the grids, with the grid ID representing trajectory points which fall into the grid. If one or more points of a trajectory fall into a grid, then the density value of the grid will increases by one. After the trajectory mapping, we calculate the density of each grid and determine whether a grid is a frequent square according to the user-specified minimum support threshold s_{min}. If the grid density of a square is greater than s_{min}, then the square is a frequent square; otherwise, it is an infrequent square.

According to the user-specified minimum support threshold s_{min}, all frequent squares could be found. However, the number of frequent grid may be very large. To reduce the number of frequent grid, the methods of merging of frequent area formation are discussed in the following paragraphs.

Definition 1 frequent area sets: suppose the trajectory space of an object can be divided into $n\times m$ squares collections. Let ς denote the set of these $n\times m$ squares. Given the density of each grid and the minimum support threshold s_{min}, the set of the frequent areas obtained by merging the frequent squares, denoted by R, satisfies the following six conditions:

(1) Each r in the region of R ($r \in R$) is a rectangular;

(2) All r in the region of R ($r \in R$) are mutually disjoint;

(3) For each frequent square in ς , there exists an area $r \in R$ such that the frequent square belongs to the area *r*;

(4)The average density of each regions *r* ($r \in R$) is greater than s_{min}, (i.e. $avg_{(i,j)\in r}\varsigma(i,j) \geq s_{\min}$);

(5) Assuming the set of a region *r* is $h\times k$, all the superset $r'(r' \supseteq r)$ of size (*h*+1)

$\times k$ or $h \times (k+1)$ do not satisfy condition (4) (i.e. R and r' contain the same frequent square.);

(6) The boundary function B(m, n) satisfies the condition: $h \leqslant m$, $k \leqslant n$。

By definition 1, in order to ensure that the final frequent area obtained through merging is a rectangle, the area could only expand in four directions: up, down, left, right. In the process of merging, we introduce the concept of boundary function to limit the maximum area space after the merging of squares.

$$B(m,n) = \{x_{\max} - x_{\min} \le m, y_{\max} - y_{\min} \le n\} \tag{1}$$

Formula (1) gives the definition of the boundary function. Given the area obtained through merging is a rectangle; the maximum number of grids area could possibly contain is $n \times m$, as n and m confine maximum area of one rectangle. As a grid is denoted by (i, j), the maximum values of column and row in area formed by square merging are denoted by x_{max}, y_{max} respectively; while the minimum values are denoted by x_{min}, y_{min}, respectively.

According to the condition of merging, it is only possible to expand the middle six squares (light color area) in four directions (red area and blue area), as is shown in Figure 2. If the middle six squares is to expand in one specific direction, the area must satisfy the conditions of the boundary function in the first place. For example, suppose the boundary function is set to be B (3, 3), then the region could only expand in two directions (blue area): either up or down.

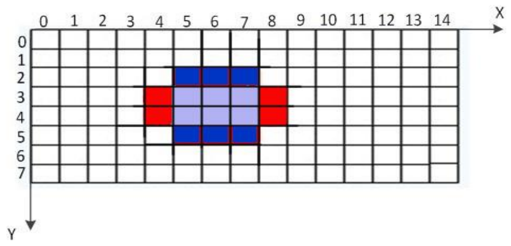


Figure 2. Bound Function

In addition, it is highlighted that the average support value of the region which is formed through square merging, rather than the density of the whole region, is considered in the process of region merging. Besides, the merging is conducted only if the new area contains frequent squares. The region merging algorithm is as follows.

2.3. Region Merging Algorithm

Algorithm: MergeRegion (ς , s_{min})

Input: Square set ς , Density of each square is ς (i,j), Minimum support threshold s_{min}

Output: Set R of frequent area.

```
1. R=$\varnothing$ ; $\varsigma^* = \{(i, j) \in \varsigma \mid \varsigma(i, j) \geq s_{min}\}$
2 foreach $(i, j) \in \varsigma$ do $used(i, j) = false$
3 foreach $(i, j) \in \varsigma^*$
4   if $\neg used(i, j)$
5     $r = \{(i, j)\}$
6    repeat
7      foreach $dir = \{left, right, up, down\}$ do
8        $r_{dir}$ = r extended on direction dir;
9        $ext = \{dir \mid r_{dir} \in \varsigma \wedge avg\_density(r_{dir}) \geq s_{min}$
              $\wedge \exists(h,k) \in (r_{dir} \setminus r).\varsigma(h,k) \geq s_{min}$
              $\wedge \forall(h,k) \in r_{dir}.\neg used(h,k)$
              $\wedge (max(h) - min(h)) \leq m$
              $\wedge (max(k) - min(k)) \leq n\};$

10         if $ext \neq \varnothing$ then
11           $dir = arg\ max_{d \in ext} avg\_density(r_d)$
12           $r = r_{dir}$
13      until $ext = \varnothing$
14      foreach $(i, j) \in r$ do $used(i, j) = true$
15        $R = R \bigcup \{r\}$
16 return R;
```

The algorithm takes each frequent into consideration recursively (step 1: filter the infre- quent grid; step 3, 4: consider the frequent grids which are not covered by any frequent area). It creates a set for every frequent grid which satisfies above-mentioned conditions (step 5), and attempts to expand in the four directions (step 7, 8). Among the four, all conditions which satisfy definition are found (step 9). Then, the average support values of the region formed after expanding in each direction are calculated and compared with one another. Lastly, the region expands in the direction with the largest average support value (step 10-12).

3. The Experimental Results and Analysis

The method proposed in this paper is testing using actual data containing 145 trajectories from two school buses in downtown Athens [12], with an acquisition frequency of 30s. Figure 3(a) shows the frequent area obtained by the method from literature [10]. It is noticed from the figure that although some regions occupy a relatively large space, it appears in the transformed sequence in the form of a point, as one region only represents. As a result, the movement patterns of a moving object in the large areas are lost, which further leads to a loss in frequent trajectories. Figure 3(b) indicates the frequent area obtained using the method proposed in this paper, where a large frequent area is divided into multiple small frequent areas. By comparing the two figures, it is evident that the maximum size of an area is effectively limited by the boundary function defined, which allows us to conduct a more comprehensive analysis on movement patterns of a given object. Certainly, the value of *m* and *n* in the boundary function may vary in different applications.

In addition, it should be noticed that the side length of a grid *d* = 200 is not an optimal value, just an experimental data. In the process of frequent trajectory mining which will be discussed later, we will find out the optimal of *d* through experimental analysis and comparison.

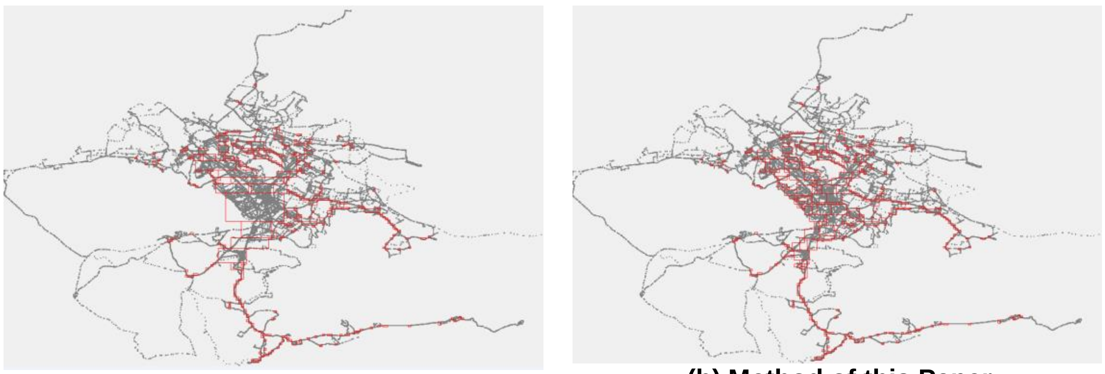

(a) Method of Literature [10] **(b) Method of this Paper**

Figure 3. Frequent Regions（s_{min}=10，*d*=200，*m*=5，*n*=5）

In order to determine the optimal value of *d* (side length of a grid), two trajectories are selected randomly from the set of experimental data for testing. As is shown in Figure 4, if the minimum support threshold is s_{min}=2, then the results of the frequent trajectory mining process are expected to be T_a and T_b shown in the dotted box.

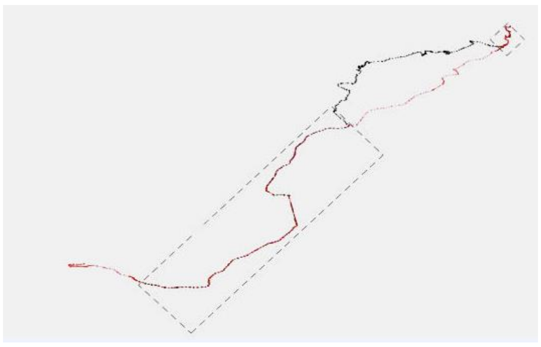
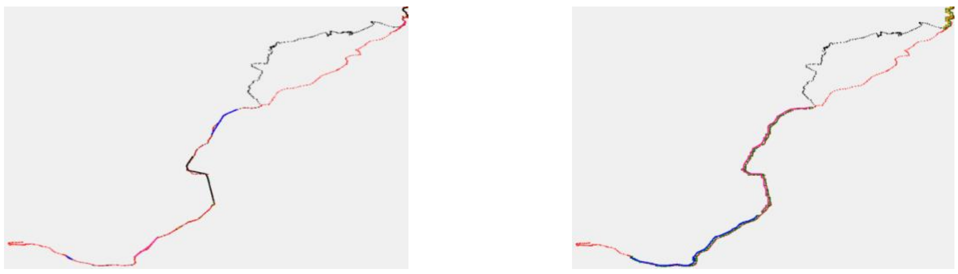
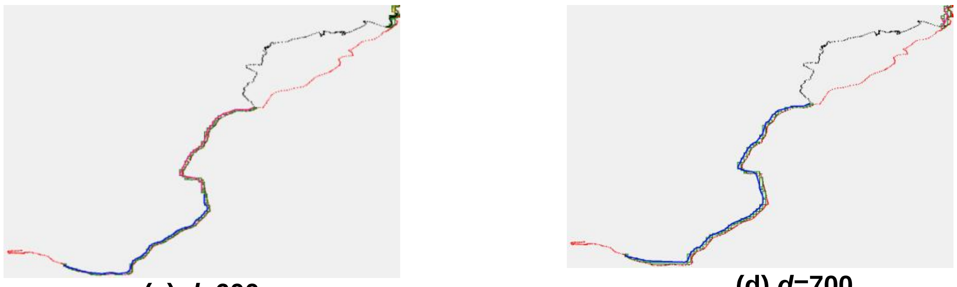


Figure 4. Two Trajectories

Figure 5 indicates various results of frequent trajectory mining process when grid size *d* is different. According to the figure, though two expected frequent trajectories are obtained wh- en *d* = 700, the result deviates from the object's original trajectory, thereby it failing to satisfy the requirements. The expected results could also be obtained when *d*= 500 and *d*= 600, yet a complete trajectory T_a is decomposed into multiple sub-trajectories. Therefore in this paper, *d* is set to be 500 and further experiments are conducted to determine τ , in order to obtain a desirable outcome.


(a) *d*=100 **(b) *d=500***

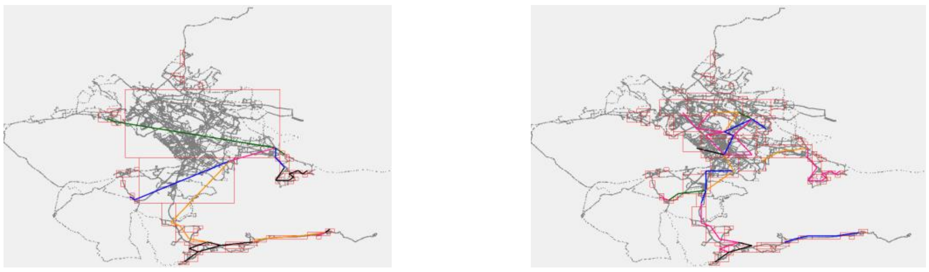


(c) *d*=600 (d) *d*=700

Figure 5. Frequent Trajectories

The value of boundary function B(*m*, *n*) in trajectory data clustering algorithm are determined according to specific situations(different mining granularity) and applications. For example, B(*m*, *n*) should be of a size of a few kilometers if the frequent trajectories of a moving object we are to acquire are within a city. This value could change to approximately the size of a county if the object is within a province. Since the experimental data in this paper comes from school buses in Athens, we set *m*=*n*= 5, where *m* and *n* are parameters of boundary function B(*m*, *n*). In other words, the maximum area of frequent area cannot exceed $md \times nd$ ($5d \times 5d$).

Figure 6 shows frequent trajectories obtained by using two different trajectory data clustering method. It is observed that the method proposed in literature [10] leads to great loss in frequent trajectories in large areas, while two parameters *d* and B(*m*, *n*) proposed in this paper may control the maximum frequent area effectively, thereby avoid the frequent trajectory loss in large areas.



(a) Poular Region (b) Dynamic Bound

Figure 6. Frequent Trajectories (s_{min}=10)

Table 1 shows the effect of different data gathered.

Table 1. Clustering Result

Data	Method of literature10	Method of this paper
19950	39 classes	45 classes
31867	94 classes	104 classes
45080	184 classes	211 classes
56261	195 classes	239 classes
59619	225 classes	269 classes
66096	248 classes	296 classes

It can be seen from Table 1, the method can effectively control the largest space of frequent area for different data sets, thus ensures frequent trajectories in large area will not be lost.

4. Conclusion

This paper puts forward a trajectory data clustering method based on dynamic grid division-merging, in order to address problems existing in the current method which is based on grid division, such as difficulties in grid size determination, and trajectory loss in large areas. In this method, grid size is set as a limiting parameter, and is determined by experimental analysis. Besides, the maximum space of trajectory area formed by grid merging is restricted by boundary function B (m, n). Experiments show that this method could resolve disadvantages of trajectory data clustering method based on grid partitioning and performs higher flexibility and adaptability.

Acknowledgments

This work was supported by a grant from the Natural Science Foundation of China (No. 61172018), the Science & Research Plan Project of Xi'an City(No.CXY1340(2)).The authors are grateful for the anonymous reviewers who made constructive comments.

References

[1] W. Hu, X. Li, G. Tian, S. J. Maybank and Z. Zhang, "An Incremental DPMM-Based Method for Trajectory Clustering, Modeling and Retrieval", IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 5, **(2013)**, pp. 1051-1065.

[2] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao and D. W. Cheung, "Mining, indexing, and querying historical spatiotemporal data", In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04), ACM, New York, NY, USA, **(2004)**, pp. 236-245.

[3] G. Costa, G. Manco and E. Masciari, "Effectively grouping trajectory streams", New Frontiers in Mining Complex Patterns. Springer Berlin Heidelberg, **(2013)**, pp. 94-108.

[4] I. Tsoukatos and D. Gunopulos, "Efficient mining of spatiotemporal patterns", Advances in Spatial and Temporal Databases, **(2001)**, pp. 425-442.

[5] J. Kang and H.–S. Yong, "Mining trajectory patterns by incorporating temporal properties", Proceedings of the 1st International Conference on Emerging Databases, **(2009)**, pp. 63–68.

[6] J. G. Lee, J. Han and K. Y. Whang, "Trajectory clustering: a partition-and-group framework", Proceedings of the 2007 ACM SIGMOD international conference on Management of data. ACM, **(2007)**, pp. 593-604.

[7] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH: an efficient data clustering method for very large databases", ACM SIGMOD Record. ACM, vol. 25, no. 2, **(1996)**, pp. 103-114.

[8] J. Kang and H.–S. Yong, "Mining spatio-temporal patterns in trajectory data", Journal of Information Processing Systems, vol. 6, no. 4, **(2010)**, pp. 521-536.

[9] N. S. Savage, S. Nishimura, N. E. Chavez, and X. Yan, "Frequent trajectory mining on GPS data", In Proceedings of the 3rd International Workshop on Location and the Web (LocWeb '10), Erik Wilde, Susanne Boll and Johannes Schöning (Eds.). ACM, New York, NY, USA, **(2010)**, pp. 1-4.

[10] F. Giannotti, M. Nanni, F. Pinelli and D. Pedreschi, "Trajectory pattern mining", Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, **(2007)**, pp. 330-339.

[11] L. Chen, M. Lv, Q. Ye, G. Chen and J. Woodward, "A personal route prediction system based on trajectory data mining", Information Sciences, vol. 181, no. 7, **(2011)**, pp. 1264-1284.

[12] http://www.chorochronos.org.