An Approach of Ship Trajectory Clustering Based on Minimum Bounding Rectangle and Buffer Similarity

Mengzhu Sun\textsuperscript{1,2} and Jiasheng Wang\textsuperscript{1,*}

\textsuperscript{1}Faculty of Geography, Yunnan Normal University, Kunming, China
\textsuperscript{2}School of Management Science and Engineering, Baise University, Baise, China

*Corresponding author: jersonwang@ynnu.edu.cn

Abstract. A clustering method based on Minimum Bounding Rectangle and buffer similarity was proposed to improve the trajectories clustering and reduce computation time. Taking the Volunteer Observation Ship data as an example, firstly, we calculated similarity by dividing the area of the overlapped parts of two MBRs by the area of the larger MBR. According to the similarity matrix, the original trajectories were divided into several clusters by coarse clustering using Density-Based Spatial Clustering of Applications with Noise (DBSCAN). After that, the similarity of each cluster was calculated by buffer analysis. Then DBSCAN was used again for each cluster, and the Silhouette Coefficient was used to evaluate the resulting cluster quality. The experimental results showed that the presented methods improve the accuracy of MBR and reduce time cost of buffer similarity.

Keywords: Ship Trajectory, Clustering, Minimum Bounding, Buffer Similarity.

1. Introduction

With the rapid development of satellite positioning technology and mobile communication technology, many motion trajectory data increased including flight, vessel, vehicle, and pedestrian. The static and dynamic information of moving object are contained in these trajectories and the behavior patterns hidden behind the data can be found by analyzing the trajectories, which could provide reference for user service promotion and public safety management. At present, trajectory data analysis has been used in many fields, such as traffic safety [1], animal migration [2], and meteorological monitoring [3]. Clustering analysis is one of the most important approaches of trajectory data analysis which divides trajectories into different groups according to the similarity to reflect the spatial and temporal patterns of trajectories [4]. Distance/similarity calculation and clustering method are the core steps of clustering analysis.

Extensive research has been completed on the method of trajectory distance/similarity calculation [5]. The distance/ similarity calculation could be divided into shape-based distance/similarity and temporal-based distance/similarity depending on whether temporal characteristics are considered. Temporal-based distance/similarity calculation takes time factor into account, such as Euclidean Distance [6], Dynamic Time Warping (DTW) [7], Longest Common Sub-Sequence (LCSS) [8], and Edit Distance [9]. The shape-based trajectory distance/similarity regard a trajectory as a whole, regardless of the time factor, and calculates the distance/similarity according to the position of trajectory...
points. Hausdorff distance [10] and Fréchet distance [11] are widely used in various fields, but none of them regards the trajectory as a whole. L. Bin et al [12] proposed a new method of distance measurement named One Way Distance (OWD), which measures distance using the average of distance from each point of one trajectory to the nearest point of the other trajectory. Based on the characteristics of the above methods, Philippe C [13] introduced a new distance referred to as the Symmetrized Segment-Path Distance (SSPD), which is the average distance of the minimum distance from each point of one trajectory to the nearest point of the other trajectory. OWD is based on the point-point distance, considering the shape and position of the trajectory while SSPD considered point-path distance. However, the results of these methods are affected by the intensity of the distribution of trajectory points.

The distance based on the geometry is not affected by the intensity of the trajectory point distribution because it considers the trajectory as a whole. W. Jiasheng [14] proposed a method for calculating the trajectory distance based on buffer analysis, which use the ratio of buffer intersection area to the larger buffer area to denote the trajectory distance. It is more accurate to call this distance as “dissimilarity”, where its value range from 0-1, and is independent of the intensity of the trajectory point distribution. It is necessary to explore a method that can ensure the stability of the distance value and the computational complexity because the high computation cost involves the calculation of buffer and area.

Clustering methods have been relatively mature, including five categories: partition- based [15], hierarchy-based [16], density-based [17], grid-based, and model-based. Density-based methods can quickly find the clusters (DBSCAN) of any shape and easily identify noise in the data, which meets the requirement of trajectory clustering. Much research has been done on density-based methods to achieve trajectory clustering [18, 19].

Overall, buffer analysis is not affected by the trajectory distribution, but it has high computation cost for massive trajectories. Therefore, a progressive clustering based on MBR and buffer is proposed to improve the computational efficiency of trajectory clustering based on buffer analysis. Firstly, the first coarse clustering is carried out according to the similarity matrix calculated by Minimum Bounding Rectangle (MBR). Then, DBSCAN is used for each cluster according to the similarity matrix of each cluster calculated by buffer analysis.

### 2. Density Clustering Based on MBR and Buffer Analysis

#### 2.1. Method of Trajectory Distance Calculation

**2.1.1. MBR Distance.** MBR is the minimum rectangle determined by the range of horizontal and vertical value of the trajectories. MBR of trajectory is a rectangle formed by the range of abscissa \([x_{min}, x_{max}]\) and the range of ordinate \([y_{min}, y_{max}]\) of trajectory. The vertex coordinates of MBR are \((x_{min}, y_{max}), (x_{max}, y_{max}), (x_{max}, y_{min}), (x_{min}, y_{min})\). Among them, \(x_{min}\) is the minimum of abscissa, \(x_{max}\) is the maximum of abscissa, \(y_{min}\) is the minimum of ordinate and \(y_{max}\) is the maximum of ordinate.

The intersecting area of two MBRs \((A_i)\) and the area of the larger MBR \((A_{max})\) are used to calculate MBR Distance (MBRD), with a value ranging from 0-1. The MBRD is \((1-A/A_{max})\) when the MBRs of two trajectories intersect, and the MBRD is one in other case.

**2.1.2. Buffer Distance.** A buffer distance references the definition in W. Jiasheng [14]. Trajectory buffer calculation is the first step in the calculation of the buffer distance. A buffer is a polygon, which is within a certain distance to a point, line, or polygon. It refers to an impact area or service area of a spatial object. The basic idea of buffer analysis is to confirm the neighborhood of a given spatial object or set of objects, and the size of the neighborhood is determined by the neighborhood radius, \(R\). Therefore, the buffer of object \(O_i\) is defined as:

\[
B_i = \{x:d(x,O_i) \leq R\}
\]  

Equation (2) defines the buffer of object \(O_i\), and the buffer is a set of points in which the distance (d) between the point and \(O_i\) is less than \(R\). In addition, d usually takes the Euclidean distance.
Buffer distance is calculated as the ratio of the intersection area of the buffer to the area of the larger buffer. If \( A_{r1} \) and \( A_{r2} \) are the area of the two buffer polygons, \( A_{r3} \) is the intersection area of the buffers, and the buffer distance can be expressed as:

\[
S = \frac{A_{r3}}{\max(A_{r1}, A_{r2})}
\]

(2)

The buffer of trajectory can comprehensively reflect the spatial features such as the position, length and the shape of trajectory. Therefore, the buffer distance of the two trajectories can perform the similarity commendably.

2.2. DBSCAN

DBSCAN defines clusters as the largest set of points that are density-connected. It can divide a region with high density into clusters, and it can find clusters in any shape in a noisy spatial database. There are two main parameters: \( \epsilon \) and \( \text{MinPts} \), where \( \epsilon \) is the neighbor parameter, and \( \text{MinPts} \) is the minimum number of points in \( \epsilon \). DBSCAN can described as: starting from a selected core point, continuously expanding to a density reachable area to obtain a maximal area containing core points and boundary points, where the density of any two points is connected. For the data set \( X = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \), the goal of DBSCAN is to divide set \( X \) into \( K \) clusters (\( K \) is also obtained by the algorithm without prior specification) and the set of noises points. To do this, the array of cluster label is introduced.

\[
m_i = \begin{cases} 
1 & (j > 0), x^{(i)} \text{ belongs to } j \text{ cluster } \\
-1 & x^{(i)} \text{ is the noise point}
\end{cases}
\]

(3)

Thus, the goal of DBSCAN is to generate an array of markers \( m_i, i = 1, 2, \ldots, N \), and \( K \) is the number of mutually non-negative numbers in \( \{m_i\}_{i=1}^N \).

2.3. Clustering Performance Evaluation Method

As an unsupervised learning task, it is necessary to evaluate the effect of clustering. Otherwise, the results of clustering would be difficult to apply. If the ground truth labels are not known, Silhouette Coefficient (SC) and Calinski-Harabaz Index (CHI) are two usual evaluation method of clustering performance.

SC is an evaluation method of clustering performance, that was proposed for the first time by Peter J. Rousseeuw in 1986 [20]. It combines two factors of cohesion and separating, calculating the SC of each sample first, and then the average of the SC of all samples. It can be used to evaluate the impact of different algorithms or algorithms with different operating modes on the clustering results based on the original data. Then the SC \( S(i) \) for a single sample is given as:

\[
S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}
\]

(4)

Where \( a(i) \) is the average distance from sample \( i \) to the other samples in the same cluster, \( b(i) \) is the average distance from sample \( i \) to the samples of other clusters.

CHI describes compactness and separation by the within-cluster dispersion and between-clusters dispersion. The formula is as follows:

\[
\sigma(k) = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N-k}{k-1}
\]

(5)

Where \( N \) is the number of training samples, \( k \) is the number of clusters, \( B_k \) is the between group dispersion matrix, \( W_k \) is the within-cluster dispersion matrix. \( B_k \) and \( W_k \) defined by:
With $C_q$ is the set of points in cluster, $c_q$ is the center of cluster $q$, $c$ is the center of $E$, $n_q$ is the number points in cluster $q$.

2.4. Framework of Progressive Clustering
In this paper, according to the MBR Similarity, DBSCAN was used for coarse clustering of trajectory. Then, using the buffer distance combined with the DBSCAN, the result of each cluster in the coarse clustering was meticulously cluster again. The final clustering results were obtained by combining the results of coarse clustering and meticulous clustering. The method was implemented using Python and Scikit-Learn and preferences for parameters were determined by clustering performance evaluation. The specific implementation process of this paper includes the following steps:

a. The MBR distance of two trajectories is calculated by calculating the MBR of each trajectory and the MBR distance matrix is finally obtained.

b. The coarse clustering was performed according to the MBR distance matrix.

c. The buffer analysis method is used to separately calculate the buffer distance of each trajectory in the coarse cluster and generate a corresponding distance matrix.

d. Meticulous clustering was performed according to the buffer distance matrix.

3. Experiments and Results

3.1. DataSets
The data used in this study is Volunteer Observation Ship (VOS) data from 2012 to 2016. It provides information about the position of ships at different times during navigation, including navigational speed, wind speed, wind direction, air temperature, air pressure, and sea surface temperature. In this paper, we mainly used ship name and location information. The data is collected from the National Climatic Data Center (https://www.ncdc.noaa.gov/data-access/marineocean-data/vosclim) of the National Oceanic and Atmospheric Administration (NOAA). Classifying the tracking point by ship's name and then 2651 trajectories were generated according to time and location information. All algorithms in this paper were implemented by Python and performed in the stand-alone computer whose CPU is Intel(R) Core (TM) i5-2450M CPU @ 2.50 GHz and internal storage of 4 GB.

3.2. Analysis of Coarse Clustering Results Based on MBR
Calculating MBR distance and generating the similarity matrix of MBR according to algorithm 1 in the third part. Then visualizing the matrix $2651 \times 2651$, as shown in Figure 1, we can see that the similarity of most trajectories is close to 1. In addition, there are plenty of trajectories whose similarity is 0, but there are fewer than the number of the trajectories with similarity close to 1. The specific quantity can be further obtained by clustering.
Figure 1. Visualization of MBR similarity matrix

DBSCAN is used to carry out coarse clustering of trajectories according to the calculated similarity matrix of MBR. The coarse clustering involves the preferences for parameters in density clustering, including eps and MinPts, which are selected by SC and CHI. During the process, one parameter is fixed and another parameter is set to calculate SC and CHI.

We first set the value of the MinPts to 5, then change the value of the eps. SC and CHI are calculated until they reach to reasonable values. Afterwards, the values of MinPts were set to 10, 15, and 20, constantly changing the value of eps to calculate the SC and CHI separately. Figure 2 shows the calculated SC and the CHI in different parameter settings.

Figure 2. The SC and CHI of different preferences for parameters in coarse clustering

SC and CHI were integrated to select the parameters of DBSCAN cluster. In principle, a larger SC, indicates a better cluster performance. However, it is necessary to consider the real clustering results when selecting clustering parameters. Although SC increased with the increase of eps, the performance of clustering was not ideal when the value of eps was maximized. If the value of eps is high, there were
only a few clusters, with the final clustering results being only one or two. The result of CHI is like the result of real clustering. A higher CHI value indicates more effective clustering. In the end, combined with the value of SC and CHI, the values of eps were calculated when MinPts = 5, 10, 15 and 20. The values of SC and CHI are shown in Table 1.

Table 1. Statistics of different parameter settings of coarse clustering

| eps  | MinPts | Number of clusters | Number of outliers | Number of maximum clustering trajectories | SC      | CHI    |
|------|--------|--------------------|--------------------|------------------------------------------|---------|--------|
| 0.18 | 5      | 16                 | 506                | 1617                                     | 0.058   | 357.131|
| 0.22 | 10     | 7                  | 508                | 1680                                     | 0.293   | 818.509|
| 0.18 | 15     | 7                  | 806                | 1480                                     | 0.129   | 781.052|
| 0.2  | 20     | 6                  | 800                | 1518                                     | 0.255   | 929.496|

Likewise, the real clustering effect and the value of SC and CHI should be considered when the clustering parameters are finally determined. The clustering results under different parameter settings are shown in Figure 3. Combined with Table 1, eps = 0.2 and MinPts = 20 were selected as the final parameters for coarse clustering. After eliminating 800 outlier trajectories, the set of trajectories is clustered into six clusters, with the largest cluster having 1518 trajectories, which is consistent with the result of similarity matrix visualization. Coarse clustering results need to be refined to further analyze the information contained in the trajectory data set.

Figure 3. The first clustering results with different parameters setting

3.3. Analysis of Meticulous Clustering Results Based on Buffer Analysis

The results of first clustering were divided into six classes: A, B, C, D, E and F. The buffer analysis method was used to calculate the similarity of each class. It’s crucial for the method to calculate the buffer distance. We calculated the value of buffer distance as 0.4 by the density of grid. Similarity calculation considers not only the accuracy of the calculation, but also the time cost in the calculation. Table 2 shows the time cost in the calculation of buffer similarity of each class.

Table 2. Time cost statistics for similarity calculation of different class

| Class | Number of trajectory | Buffer distance | Time cost |
|-------|----------------------|-----------------|-----------|
| classA| 1680                 | 0.4°            | 2271.40s  |
| classB| 90                   | 0.4°            | 134.51s   |
| classC| 51                   | 0.4°            | 118.37s   |
| classD| 74                   | 0.4°            | 140.28s   |
| classE| 100                  | 0.4°            | 123.40s   |
| classF| 18                   | 0.4°            | 121.99s   |
The same method as above was used for the parameters’ selection. The final parameters are set as shown in Table 3. Averaging the SC of all classes, the value is 0.354, which is greater than the value in first clustering which is 0.255. This shows that our refined clustering method is more effective than the original single method. The result calculated by the buffer similarity is better than calculated by MBR. However, the speed of buffer similarity calculation is slow. We used the buffer analysis method to calculate the similarity of the set of original trajectories, which takes 7531.13s, about two hours. It could greatly reduce the computation time by integrate two similarity calculation methods. In summary, combining of the two methods can improve the computational efficiency, at the same time, the accuracy of clustering results is improved.

### Table 3. Statistics of parameter settings and clustering results in meticulous clustering

| Classes | eps | MinPts | Number of clusters | Number of outliers | SC   | CHI   |
|---------|-----|--------|--------------------|--------------------|------|-------|
| class A | 0.21| 10     | 10                 | 454                | 0.035| 209.252|
| class B | 0.23| 5      | 1                  | 4                  | 0.461| 35.131|
| class C | 0.23| 5      | 2                  | 5                  | 0.487| 86.155|
| class D | 0.24| 5      | 3                  | 13                 | 0.366| 62.931|
| class E | 0.11| 10     | 1                  | 21                 | 0.438| 71.123|
| class F | 0.17| 10     | 1                  | 11                 | 0.336| 37.494|

![Figure 4](image_url)  
**Figure 4.** Diagram of clustering results. (a) Clustering results for different parameter sets of class A; (b) Original set of trajectories; (c) Clustering results of our method; (d) Clustering results of buffer analysis.

For the clustering results of class A, we tried to set different parameters, and calculate the values of SC and CHI as shown in Table 4. Figure 4(a) is the clustering result of class A according to the order
parameters in Table 4. Table 4 and Figure 4(a) show that although the value of SC and CHI are higher under other parameter settings, the clustering effect is not ideal, and the larger clusters are not clustered out. Integrated the visual interaction results, the final clustering parameters of class A class are \( \text{eps} = 0.21 \) and \( \text{MinPts} = 10 \).

The final clustering result is shown in Figure 4(c). Compared with the trajectories before clustering, the trend and direction of the trajectories are more obvious after repeated clustering. After eliminating the redundant trajectories, the trajectories were clustered into 17 clusters, from which the general directions of the routes can be derived. Compared with the previous trajectories without clustering, the trajectories are clearer, but there are also shortcomings, some routes are not clustered out and eliminated as redundant routes. In addition, the results of clustering are not good for different lengths of trajectories.

| \( \text{eps} \) | \( \text{MinPts} \) | Number of outliers | Number of clusters | SC    | CHI   |
|-----|------|------------------|-----------------|------|------|
| 0.29 | 5    | 125              | 5               | 0.173| 375.400|
| 0.32 | 5    | 105              | 2               | 0.291| 731.543|
| 0.21 | 10   | 454              | 10              | 0.035| 209.252|
| 0.23 | 10   | 348              | 7               | 0.047| 280.551|
| 0.24 | 10   | 287              | 6               | 0.035| 308.368|
| 0.25 | 10   | 259              | 4               | 0.167| 469.897|
| 0.28 | 10   | 172              | 3               | 0.224| 613.619|
| 0.29 | 10   | 167              | 3               | 0.229| 624.231|
| 0.31 | 10   | 136              | 2               | 0.282| 750.392|

3.4. Clustering Results Comparison between Our Method and Buffer Analysis

It has high computational complexity of the similarity calculation based on buffer distance. Computing the experimental similarity takes 7531.13 s, and our method combined the similarity calculation of MBR and buffer analysis, improved the calculation efficiency of trajectory clustering based on buffer analysis. The calculation time is only 2909.95 s, reduced by 4621.18 s, which indicates that our method improves the calculation efficiency. The similarity calculated by buffer analysis is clustered by DBSCAN. Compared with the results of the proposed method, the clustering effect of our method is better than buffer analysis (as shown in Figure 4(d)).

4. Conclusions

A progressive clustering method based on similarity calculation of MBR and buffer analysis was proposed in this paper, which reduces the calculation time and effectively improves the accuracy of clustering. The combination of SC and CHI were used to evaluate the clustering effect. The MBR similarity calculation has a high computation rate, but the clustering effect is not very good. Although the buffer analysis has high accuracy, the computation rate is too slow. These two methods are complementary and could be combined.

Setting value of buffer distance is the key of similarity calculation by buffer analysis. For the large clusters, the clustering results are not ideal when the clustering evaluation parameters are large. Future studies should examine techniques that improve the accuracy of clustering.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (No. 41501436).

References

[1] W. Jiasheng, L. Manchun, L. Yongxue, et al, Safety assessment of shipping routes in the South China Sea based on the fuzzy analytic hierarchy process, Safety. Sci. 62 (2014) 46-57.
[2] W. Yuwei, L. Ze, T. John, et al, A new method for discovering behavior patterns among animal
movements, Int. J. Geogr. Inf. Sci. 30 (2015) 929-947.
[3] J.N Cape, J. Methven, L.E. Hudson, The use of trajectory cluster analysis to interpret trace gas measurements at Mace Head, Ireland, Atmos. Environ. 34 (2000) 3651-3663.
[4] G. Xi, P. Tao, S. Jia, L. Ming, Review of the Research Progresses in Trajectory Clustering Methods, Progress in Geography, 30 (2011) 522-534.
[5] T. Górecki, Using derivatives in a longest common subsequence dissimilarity measure for time series classification, Pattern Recogn. Let. 45 (2014) 99-105.
[6] Michael R. Berthold, F. Höppner, On Clustering Time Series Using Euclidean Distance and Pearson Correlation, Expert. Syst. Appl. 52 (2016) 26-38.
[7] D. J. Bemdt, J. Clifford, Using dynamic time warping to find patterns in time series, Kdd Workshop. (1994) 359-370.
[8] R. Agrawal, K.-I. Lin, H. S Sawhney, et al, Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases, in International Conference on Very Large Data Bases, 1995, pp.490-501.
[9] T. Bozkaya, N. Yazdani, M. Özsoyoğlu, Matching and indexing sequences of different lengths, Proc. of the Cikm. Las. Vegas. (1997) 128-135.
[10] D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, Comparing Images Using the Hausdorff Distance, IEEE T. Pattern. Anal. 15(1993) 850-863.
[11] H. Alt, M. Godau, Computing the fréchet distance between two polygonal curves, Int. J. Comput. Geom. Ap. 5(1995) 75-91.
[12] L. Bin; S. Jianwen, Shapes based trajectory queries for moving objects, in ACM International Workshop on Geographic Information Systems, Bremen, Germany, 2005, pp.21-30.
[13] P.C. Besse, B. Guillouet, J.-M. Loubes, et al, Review and Perspective for Distance-Based Clustering of Vehicle Trajectories, IEEE T. Intell. Transp. 17(2016) 3306-3317.
[14] W. Jiasheng, Research on spatial comprehensive evaluation of shipping lanes in the South China Sea. Nanjing University, Nanjing, China, 2014.
[15] M. Shindler, A. Wong, A. Meyerson, Fast and accurate k-means for large datasets, in International Conference on Neural Information Processing Systems, Granada, Spain, 2011, pp.2375-2383.
[16] W. Jiasheng, L. Yongxue, Y. Kun, et al, Extraction Method of Main Routes in South China Sea Based on Spatial Cluster Analysis, Journal of Traffic and Transportation Engineering, 16(2016) 91-98.
[17] W. Di, D. Yunyan, Y. Jiawei, et al, Density-Based Spatiotemporal Clustering Analysis of Trajectories, Journal of Geo-information Science, 17(2015) 1162-1171.
[18] L. Bo, N. de S. Erica, S. Matwin, et al, Knowledge-based Clustering of Ship Trajectories Using Density-based Approach, in IEEE International Conference on Big Data, Washington, DC, USA, 2015, pp.603-608.
[19] V. Mirge, S. Gupta, K. Verma, A Novel Approach for Mining Trajectory Patterns of Moving Vehicles, Int. Journal of Com. Applications, 104 (2014) 4-8.
[20] P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1999) 53-65.
[21] T. Caliński, J Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1974) 1-27.
[22] D. L. Davies, D. W. Bouldin, A Cluster Separation Measure, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1. 2 (1979) 224-227.
[23] J. C. Dunn, Well-Separated Clusters and Optimal Fuzzy Partitions, J. Cybernetic. 4 (1974) 95-104.