Trajectory Distance Algorithm Based on Segment Transformation Distance

Longbao Wang[1], Xin Lv[1,*], and Jicun An[1]
[1]School of Computer and Information, Hohai University
Nanjing 211100, China
[e-mail: wlb@hhu.edu.cn, lvxin.gs@163.com]
[*]Corresponding author: Xin Lv

Received December 9, 2020;revised February 27, 2022; accepted March 16, 2022; published April 30, 2022

Abstract

Along with the popularity of GPS system and smart cell phone, trajectories of pedestrians or vehicles are recorded at any time. The great amount of works had been carried out in order to discover traffic paradigms or other regular patterns buried in the huge trajectory dataset. The core of the mining algorithm is how to evaluate the similarity, that is, the “distance”, between trajectories appropriately, then the mining results will be accordance to the reality. Euclidean distance is commonly used in the lots of existed algorithms to measure the similarity, however, the trend of trajectories is usually ignored during the measurement. In this paper, a novel segment transform distance (STD) algorithm is proposed, in which a rule system of line segment transformation is established. The similarity of two-line segments is quantified by the cost of line segment transformation. Further, an improvement of STD, named ST-DTW, is advanced with the use of the traditional method dynamic time warping algorithm (DTW), accelerating the speed of calculating STD. The experimental results show that the error rate of ST-DTW algorithm is 53.97%, which is lower than that of the LCSS algorithm. Besides, all the weights of factors could be adjusted dynamically, making the algorithm suitable for various kinds of applications.

Keywords: Trajectory data, Trajectory distance, Segment transform, DTW, Similarity of trajectory.

1. Introduction

In recent years, with the rapid development of technologies such as GPS and wireless communications, various mobile objects in the city can be effectively tracked. Large amounts of data describing the motion history of moving objects, known as trajectory, are currently generated and managed in scores of application domains, such as traffic condition prediction, public transportation system, video tracking, and video motion capture. Typical examples include traffic conditions and congestion levels prediction in cities by analyzing the trajectories of taxis [1], urban roads planning through analyzing sharing-bikes trajectories [2][3], urban hot spots and travel rules mining according to check-in data or bus smart card data [4], urban spatial structure and functional areas identification based on the traffic data in the area [5].

A large number of similar trajectories can help to discover the rules. The typical goal of data analysis is to cluster similar trajectories and mine patterns of movement in moving objects. Clustering analysis is the data objects are grouped, making the same group of objects have a high degree of similarity, the objects in different groups have a lower degree of similarity [6]. The criterion of the distance between trajectories is of great research significance.

There are dozens of distance measures for trajectory data in the literature. For example, there are distance measures measuring the sequence-only distance between trajectories, such as Euclidean distance and dynamic time wrapping distance (DTW). But these two methods cannot reflect the trend of the trajectory and may lose useful information easily. To solve this problem, we propose the line segment transformation distance (STD) with the idea of editing distance. Meanwhile, Segment Transformation distance of Dynamic Time Warping (ST-DTW), as an improvement of STD, is proposed with the use of DTW. The experimental results show that the proposed methods can find the similar sub-trajectory segments of the trajectory set effectively and accurately.

2. Related Work

Trajectory sequence data can be regarded as time sequence data. Many approaches of the trajectory similarity measurement are introduced from the similarity measurement to the time sequence data. Agrawal et al. proposed the Euclidean distance in 1993 and used it to measure the similarity of trajectories [7]. The Euclidean distance between trajectories requires that the trajectories have the same sampling frequency and duration, and the distance of each moment needs to be calculated, which is more sensitive to noise. In order to reduce the impact of noise, Lee et al. proposed Minimum Bounding Rectangle distances (MBR) and defined distance calculation rules for calculating the similarity between trajectories [8]. MBR can reduce the influence of noise to a certain extent, but the similar trajectories that are not at the corresponding moment will not be able to be calculated correctly.

In the process of calculating the similarity of two time sequences of the same length, Euclidean distance corresponds the trajectories at the points in same time, and the distances at each time are summed to obtain the final distance. However, in practical applications, time sequences usually shift or stretch on the time axis, and different time sequences may have different lengths. Thus, the results of the Euclidean distance calculation may be deviated or not applicable. To solve the problem which the trajectories are not similar at the corresponding time but similar after the time scaling, such as longest common subsequence algorithm (LCSS) [9], edit distance on real sequence algorithm (EDR) [10] and DTW [11] were proposed and widely applied.

Time transformation corresponds to similar algorithms relax the limitation of the time dimension, it only need to ensure the chronological order of sampling points, and do not need to have the same sampling time and sampling frequency. The LCSS was originally a distance algorithm used to calculate the text similarity, Agrawal first used it to calculate the similarity of one-dimensional time sequence. The LCSS adopted a threshold to identify the matching point pairs, so it can calculate the similarity between trajectories. The EDR algorithm is based on the editing distance, the threshold is to determine the compatibility between two points, different from LCSS, the EDR is of higher tolerance for the noise in the sequence. The earliest DTW used to identify speech, the method can be scaled in the time dimension, to recognize the same pronunciation of different speed speech purposes. Chen et al. introduced the DTW algorithm into the trajectory calculation to solve the problems of different sampling rates or time deviations of the trajectories.

For the above algorithm, the distance between two points is measured as the Euclidean distance or the Euclidean distance limited by the threshold. Two identical trajectories have different sampling frequencies, the trajectories distances according to the Euclidean distance of sampling points will have larger difference. A trajectory data is composed of a plurality of sequential sampling points, which can also be considered as composed of a plurality of directional segments. The mainstream algorithm uses the distance between each sampling point to measure the distance between trajectories. Trajectory data has the trend, but a single sampling point cannot reflect the trend of the trajectory. The directional segments can be a good reflection of the trend of the trajectory.

Lee et al. proposed a sub-trajectory clustering algorithm in 2007, which partitions a trajectory into a set of line segments, and then, groups similar line segments together into a cluster [12]. At the same time, they proposed a computational method for calculating segment distance, to be referred as the segment distance (SD). In some cases, SD cannot accurately measure the similarity between segments with his calculation rules. For example, the SD between two segments of different length is zero, when they are in the same line and one of the endpoints coincides. Trajectory segment is a segment with direction. But the SD will not change when the angle between the two segments is greater than 90 degrees. Resulting in no distinct distinction between segments.

To keep the trend of trajectory and have a better measure of the distance between trajectories, STD and ST-DTW are proposed. The STD is an algorithm to measure the distance between segments, while the Segment transformation is the process of reclosing one segment with another segment by rotation, scaling, horizontal shift and vertical shift, the STD represents the cost of segment transformation.

3. Math Trajectory Distance Algorithm Based on ST-DTW

The trajectory of mobile objects have time and space attributes. Usually, a trajectory consists of a series of points (coor, t), where coor is the coordinates of the recorded point in multidimensional space, t is the recorded time.

3.1 Segment Transformation

Definition 1 (trajectory sequence). In Euclidean space, a trajectory T that has n recorded points can be expressed as $T = \{P_1, P_2, \dots, P_n\}$.The P_i in trajectory T represents the recorded point at time t_i.

Definition 2 (trajectory segment). Trajectory segment is the directed segment consisted of two adjacent recorded points in trajectory T. The trajectory segment is expressed as $S_i = P_iP_{i+1}$, it represents the vector from P_i to P_{i+1}. That is to say, the trajectory also can be expressed as $\mathrm{T} = \{\mathrm{S}_0, \mathrm{S}_1, \ldots \mathrm{S}_{n-1}\}$.

Definition 3 (segment transformation). The process of segment transformation consists of four steps, namely rotation, scaling, horizontal shift and vertical shift. **Fig. 1** shows the four steps of segment transformation. The dotted segment R′′ is the position before conversion, the solid segment R is the position after conversion and S is the target segment.

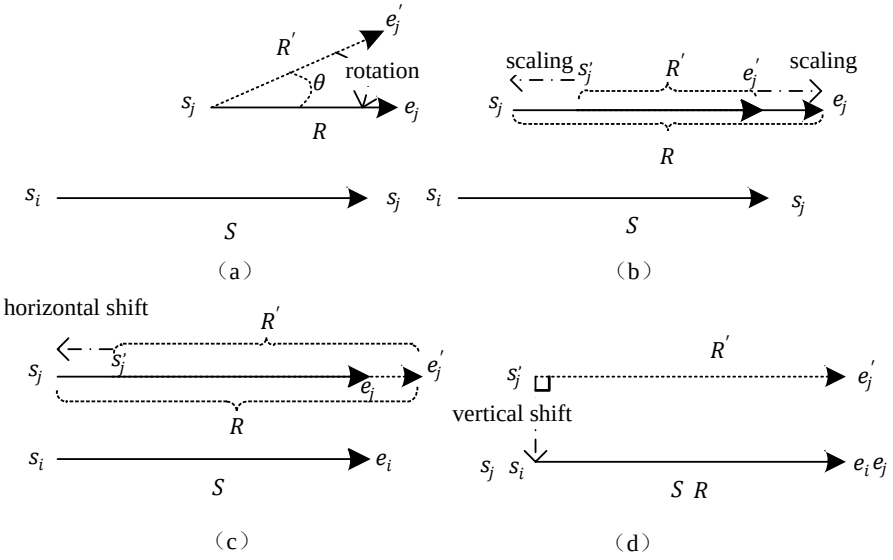

Fig. 1. Segment transformation

Definition 4 (near point). Suppose there are two trajectory segments R and S, and R is shorter than S. The point closest to trajectory segment S on the segment R is near point. When two trajectory segments intersect, the intersection point is the near point. Others, the initial point of the R is construed as the near point when the angle between two vectors in the range from 0 to 180. In (1), the θ is the angle between trajectory segment S and R, and the point s_R, e_R is the initial point and terminal point of R, the point c_{RS} is the intersection of trajectories R and S, respectively.

$$\text{near point} = \begin{cases} c_{RS} & if\ R\ intersects\ to\ S \\ s_R & if\ 0° \le \theta \le 180° \\ e_R & if\ 180° < \theta < 360° \end{cases} \tag{1}$$

Definition 5 (angle distance). The angle distance is the cost of the trajectory segment R′ rotating by θ to R along near point, where R and S have the same direction and the length of R shorter than S. The angle distance between R′ and S is defined as (2). The angle between vectors S and R′ is θ, and L_R is the length of vector R′. **Fig. 2** illustrates the angle distance through the geometry.

$$d_r = (1 - cos(\theta))L_R \tag{2}$$

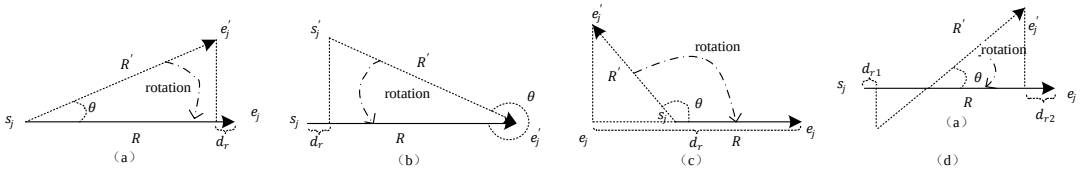

Fig. 2. Angle distance

Definition 6 (scaling distance). The scaling distance is the absolute value that one segment minus another, it defined as (3).

$$d_s = |d_R - d_S| \tag{3}$$

Definition 7 (horizontal shift distance). Suppose the trajectory segment R has the same direction and magnitude as S after rotation and scaling. The horizontal shift distance is the distance what trajectory R moves in its direction and forms a rectangle with S. **Fig. 3** shows the horizontal shift distance in geometric space, the vector R' is the position before being moved and the R is the position after being moved, d_{hs}.is the horizontal shift distance.

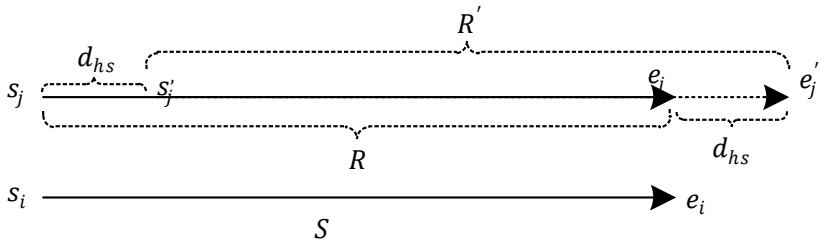

Fig. 3. Horizontal shift distance

Definition 8 (vertical shift distance). Suppose there are two trajectory segments R and S, and R is shorter than S. The vertical shift distance between R and S is the distance from near point to trajectory segment S. There are vertical shift distance with many conditions in **Fig. 4**, the value of d_{vs} is the vertical shift distance.

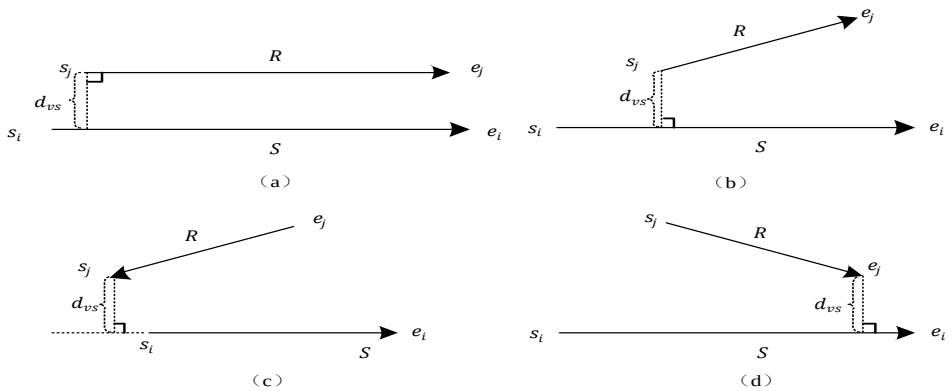

Fig. 4. Vertical shift distance

3.2 Segment Transformation Distance

In this paper, when calculating STD, the short trajectory segment rotates, stretches horizontal shift and vertical shift in turn, and the long trajectory segment remains unchanged. The transform step is from **Fig. 1(a) to Fig. 1(d)**, in the process the angle distance, scaling distance, horizontal shift distance and vertical shift distance can be calculated, and the STD is the weighted sum of four distances.

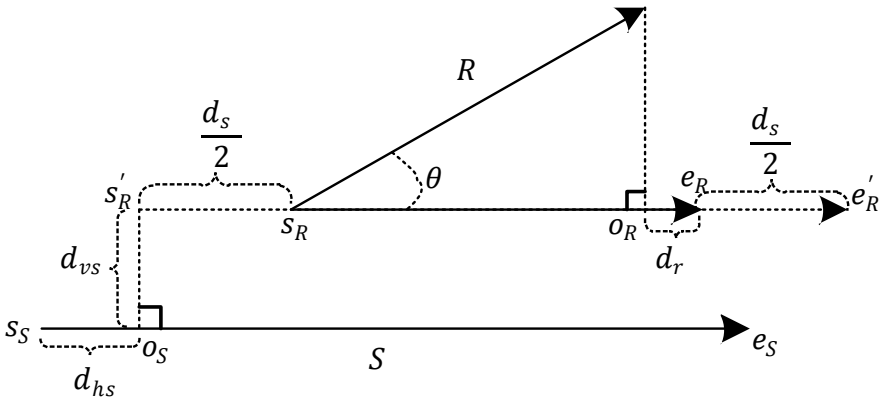

Fig. 5. Segment Transformation Distance

In **Fig. 5**, vector from s_R to o_R is parallel to $\boldsymbol{S}$, the angle between $\boldsymbol{R}$ and $\boldsymbol{S}$ is θ, the length of vector $\boldsymbol{s_R'e_R'}$ is equal to length of $\boldsymbol{S}$, the length of vector $\boldsymbol{s_Re_R}$ is equal to length of $\boldsymbol{R}$, the points o_R and o_S are pedals on the line of s_Ro_R and the line of S. (4) shows how to calculate the STD, and the $\mathrm{w_r}, w_s, w_{hs}, w_{vs}$ are the weights.

$$d_{st} = w_r d_r + w_s d_s + w_{hs} d_{hs} + w_{vs} d_{vs} \tag{4}$$

As described in Algorithm 1, the function named vector () is used to turn two points to a vector, $\mathrm{abs}(x)$ is used to get the absolute value of real number x, *calCrossoverPoint()* is used to calculate the intersection point of two segments, and *pointSegDist()* is used to calculate the distance from point to the line which the segment on. It first makes the length of the variable $\boldsymbol{R}$ less than $\boldsymbol{S}$, and calculates their vector and unit vector. Then it calculates the angle distance and scaling distance according to (2) and (3). Calculating the distances what the initial point of R to S and terminal point of R to S, then comparing them to find the near point and obtain the vertical shift distance. After being scaled, the coordinates of point can be obtained according to the near point and the unit vector of $\boldsymbol{S}$, and the horizontal shift distance will be calculated by cosine formula. Finally, according to the preset weights, the STD is obtained.

Algorithm 1 STD	
Input: Two trajectory segments $R = \{s_R, e_R\}$ and $S = \{s_S, e_S\}$	
Output: the STD between R and S	
1	$\boldsymbol{v_R}$= vector$(R), \boldsymbol{v_S}$ = vector(S) //Turn segments to vectors
2	*if* $\lvert\boldsymbol{v_R}\rvert > \lvert\boldsymbol{v_S}\rvert$ //Ensure R is a short segment
3	*then* swap$(\boldsymbol{v_R}, \boldsymbol{v_S})$
4	swap(R,S)
5	$\boldsymbol{v_{S0}} = \boldsymbol{v_S}/\lvert\boldsymbol{v_S}\rvert$ //Calculate the unit vector of long segment S

6	$d_r = \lvert \boldsymbol{v_R} \rvert - \boldsymbol{v_R} \cdot \boldsymbol{v_S} / \lvert \boldsymbol{v_S} \rvert$
7	$d_s = \lvert \boldsymbol{v_S} \rvert - \lvert \boldsymbol{v_R} \rvert$
	var d_{vs}, d_{hs}
	if R intersects to *S*
	then d_{vs}=0
	c_{rs} = calCrossoverPoint(*R*,*S*) //Calculate the intersection point
	l_{ce} = |vector(c_{rs}, e_R)|
	$e'_R = c_{rs} + l_{ce} + \boldsymbol{v_{S0}} * d_s/2$
	d_{hs} = |vector(e'_R, e_S)|
	else
8	l_s = pointSegDist(s_R, S), l_e = pointSegDist(e_R, S)
9	$d_{vs} = l_s$,
11	*if* $l_s > l_e$
12	*then* $d_{vs} = l_e$ //e_R is the near point
13	$e'_R = e_R + \boldsymbol{v_{S0}} * d_s/2$
14	$\boldsymbol{v_e}$ = vector(e'_R, e_S)
15	d_{hs} = abs($\boldsymbol{v_e} \cdot \boldsymbol{v_{S0}}$)
16	*else* $s'_R = s_R - \boldsymbol{v_{S0}} * d_s/2$ //s_R is the near point
17	$\boldsymbol{v_s}$ = vector(s'_R, s_S)
18	d_{hs} = abs($\boldsymbol{v_s} \cdot \boldsymbol{v_{S0}}$)
19	*return* $w_r d_r + w_s d_s + w_{hs} d_{hs} + w_{vs} d_{vs}$

3.3 Segment Transformation Distance of Dynamic Time Warping

The DTW algorithm is a method to represent the similarity between two sequences by calculating the optimal mapping between them through dynamic programming. The same time points of two sequences do not necessarily correspond to each other in DTW, but need to find a better correspondence through dynamic programming, so that the points in the approximate states of the two sequences correspond to each other.

Suppose there are two single-dimensional time series, $x(i), i = 1,2,\cdots,m$ and $y(j), j = 1,2,\cdots,n$, in order to calculate the DTW distances of these two sequences, a distance matrix D of $m \times n$ is first computed, where the (i^{th}, j^{th}) element is denoted as $d_{local}(i,j) = (x(i) - y(j))^2$. d_{local} denotes the local distance, which is the distance between a pair of time points in the two time sequences. In the Euclidean distance, the local distance refers to the distance between two correspondences an each same time point. In DTW, the local distance is no longer the distance between two identical time points, but can be the distance between any two time points.

W is defined as a regularization path to represent an alignment or mapping of sequences x and y.

$$W = \binom{w_x(k)}{w_y(k)}, k = 1,2,\cdots,p, \tag{5}$$

In formula (5), $w_x(k)$, $w_y(k)$ denote the subscripts of the elements in sequence x and sequence y, respectively. p denotes the length of the planning path W that satisfies $p \in [\max(m,n), m+n-1]$. $\binom{w_x(k)}{w_y(k)}$ denotes the mapping of the $w_x(k)$ element in sequence x to the $w_y(k)$ element in sequence y.

The segment transformation distance is used to calculate the distance between trajectory segments, and a trajectory has multiple trajectory segments. Therefore, the ST-DTW combined with the DTW algorithm is proposed in this paper. Different from the DTW algorithm, the ST-DTW uses the first two recorded points of trajectory to calculate the STD. The ST-DTW algorithm is calculated as follows:

$$f(\mathrm{R},\mathrm{S}) = \begin{cases} 0 & if\ m = n = 1 \\ \infty & if\ m = 1\ or\ n = 1 \\ dist_{st}\big(Head(R), Head(S)\big) + min\begin{cases} f\big(Rest(R), Rest(S)\big) \\ f(Rest(R), S) \\ f\big(R, Rest(S)\big) \end{cases} & otherwise \end{cases} \tag{6}$$

In Formula (6), m and n are the number of recorded points of trajectory R and S, $Head(R)$ is to get the first trajectory segment of R, and $Rest(R)$ is a new trajectory which is the trajectory R without the first recorded point. That is to say, $dist_{st}(Head(R), Head(S))$ represents the STD between the first trajectory segments of trajectory R and S. $f(R, S)$ is the recursive function to calculate the trajectory distance based on STD between trajectory R and S. The algorithm of ST-DWT is described as Algorithm (2).

Algorithm 2 ST-DTW	
Input: Two trajectories $R = \{p_{r0}, p_{r1}, \dots, p_{rn-1}\}, S = \{p_{s0}, p_{s1}, \dots, p_{sm-1}\}$	
Output: the ST-DTW between *R* and *S*	
1	*var distST[n-1][m-1]*
2	*for* i = 0 to n-2 // Calculate the STD between each two trajectory segments
3	*for* j = 0 to m-2
4	DistST[i][j] = STD($\{p_{ri}, p_{ri+1}\}, \{p_{sj}, p_{sj+1}\}$)
5	end
6	end
7	*var* matrix[n][m]
8	*for* i = 0 to n-1
9	matrix[i][m-1] = INF
10	end
11	*for* j = 0 to m-1
12	matrix[n-1][j] = INF
13	end
14	matrix[n-1][m-1] = 0
15	*for* i = n-2 to 0 // Calculate the ST-DTW uses dynamic programming
16	*for* j=m-2 to 0
17	matrix[i][j] = distST[i][j] + min(matrix[i+1][j],matrix[i+1][j+1],matrix[i][j+1])
18	*return* matrix[0][0]

As described in Algorithm (2), using dynamic programming instead of recursive to compute the ST-DTW. The $STD(a, b)$ is used to calculate the segment transformation distance between trajectory segment *a* and *b*, and the INF is expressed as infinity.

In general, the similarity value is between 0 and 1, and the distance value ranges from 0 to positive infinity. As the distance increases, the similarity gradually decreases. In this paper, the conversion function converts the distance to similarity with the Laplacian Kernel function.

The Laplacian kernel function reduces dependence on parameters compared to Gaussian kernel function. The conversion function is shown as (7):

$$sim(R,S) = e^{-\frac{D(R,S)}{\sigma}} \in (0,1] \tag{7}$$

Where D represents the trajectory distance calculated by ST-DTW of R and S, e is the mathematical constant and σ is the sensitivity parameter. Whit the same D, the similarity is higher when σ is larger, and the similarity is lower when σ is smaller. In **Fig. 6**, when D=1, the similarity changes with σ. When the ratio of distance D to σ is less than 0.2, the growth speed of similarity appears to be fast before it gets slow with the increasing of σ.

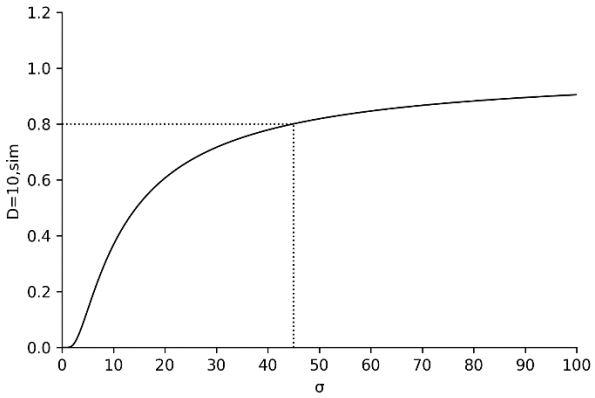

Fig. 6. The sensitivity analysis for similarity

In this paper, the ST-DTW algorithm is composed of DTW and STD, without changing the core concept of the DTW. Moreover, the STD is not only a distance between trajectory segments, but also can replace the computation method of LCSS, EDR, and other algorithms to propose new approaches to the similarity measurement.

4. Experiment Analysis

The experiments in this paper use the CVRR Trajectory Analysis Dataset from The Regents of the University of California [13] and the traffic dataset of Hefei collected through GPS equipment [14].

4.1 Dataset

The traffic dataset of Hefei contains more than 100,000 GPS trajectory data collected by volunteers' mobile phones, mainly distributed in the urban area of Hefei. **Fig. 7** shows the trajectories of the dataset. The DTW, SDTW, and ST-DTW algorithms are compared below through this dataset. Since the dataset does not give the corresponding label of the trajectory, the effect of the clustering in this algorithm is evaluated by the visual analysis. The DTW algorithm is the Dynamic Time Warping algorithm based on Euclidean distance. The SDTW is the algorithm changed the computation method of the DTW, it uses the Segment Distance proposed by Lee *et al*. in 2007. The difference between SDTW and ST-DTW is only in the computation method. Both of them use the trajectory segment for calculation. This experiment will verify the effect of segment-based calculations and point-based calculations, and the effect

of line segment distance proposed by Lee *et al*. and STD in the same scenario.

The CVRR dataset is a dataset used to evaluate trajectory clusters. All trajectories in the dataset have their own tags, so these data can be calculated according to the algorithm o get the similarity and then being clustered. The clusters will be compared with the correct class to verify the effectiveness of the algorithm. This experiment uses the CROSS dataset to verify the clustering effects of LCSS[9], EDR[10], DTW[11], SDTW[12], and ST-DTW. The CROSS dataset is shown in **Fig. 8**. The CROSS dataset simulated four way traffic intersection with various through and turn patterns present. The label of data divides the CROSS dataset into 19 clusters.

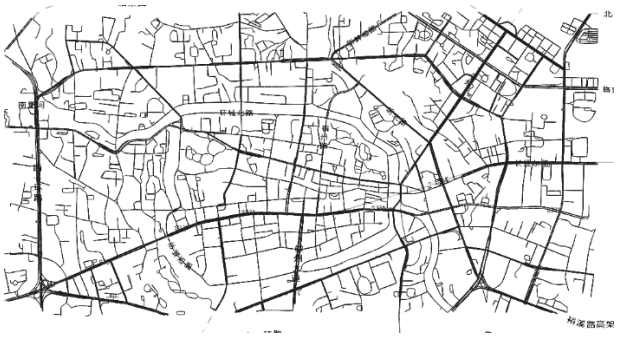

Fig. 7. Traffic dataset of Hefei

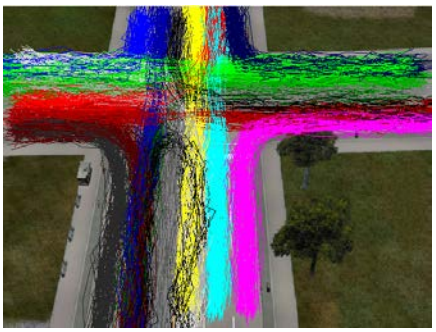

Fig. 8. CROSS dataset

4.2 Similar Trajectory Analysis

In this experiment, one trajectory data was selected, and the distances between the other trajectories were calculated using the DTW, SDTW, and ST-DTW, respectively, then select the nearest K trajectories for display. The top 25 trajectories selected by the three algorithms are shown in **Fig. 9**. **Fig. 9(a), Fig. 9(b), and Fig. 9(c)** use the DTW, SDTW, and ST-DTW algorithms to find the most similar 25 trajectories, respectively. The black curve in **Fig. 9** is the original trajectory, and the colored curves are the similar trajectories.

The similar trajectories in **Fig. 9(c)** are basically on the same main road, the **Fig. 9(a) and Fig. 9(b)** contain multiple trajectories that differ greatly from the original trajectory. It can be intuitively shown that the ST-DTW algorithm is superior to the other two algorithms through the schematic diagram.

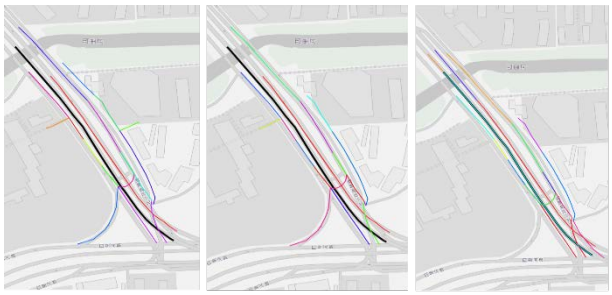

Fig. 9. Top 25 similar trajectories

4.3 Search Road Network

The ST-DTW algorithm consists of four distances, and it can be adapted to different application scenarios by setting different weights. This paper searches urban road networks by adjusting the ST-DTW weights on the Hefei traffic dataset. The ST-DTW consists of four distances: angle distance, scaling distance, horizontal shift distance and vertical shift distance. Assuming that the two trajectories are on the same road, their directions are usually in same direction or have a slight offset, and the vertical shift distance will be small. The weight of the angle distance and the vertical shift distance is increased, and the weight of the scaling distance and the horizontal shift distance are reduced. In this way, the distance between trajectories on the same road will be small. Then using the DBSCAN algorithm to cluster the trajectories, the trajectories on the same road can be clustered into the same cluster[15][16]. The clustering results are shown in **Fig. 10**. Different colors represent different clusters. Through visual analysis, it can be found that the ST-DTW algorithm with adjusting the weights can easily find the trajectories on the same road.

Fig. 10. Search road network

4.4 Clustering Effect Evaluation

In this experiment, the CVRR's CROSS dataset is used. The trajectories in this dataset already have their labels. By comparing the experimental clustering results, the effect of the algorithm can be more accurately judged. The error rate is used as an indicator to reflect the effectiveness of the algorithm. The error rate indicates the ratio of the number of trajectories which was not

clustered correctly to the total number of trajectories. The algorithm is better while the error rate is lower. The error rate is presented as follows:

$$ER = \frac{1}{N}\sum_{i=1}^{N} f(i), f(i) = \begin{cases} 0 & p_i = p_i' \\ 1 & p_i \neq p_i' \end{cases} \tag{8}$$

In formula (8), N denotes the total number of trajectories, p_i denotes the cluster belonging to the ith trajectory in the dataset, p_i' denotes the cluster calculated by algorithm, $f(i)$ is the function whether the ith trajectory is clustered correctly, 0 if correct, and 1 if clustering error.

This experiment uses CLUTO as a clustering tool to evaluate the effects of LCSS, EDR, DTW, SDTW, and ST-DTW. Resampling the dataset to make the recorded points of the dataset as uniform as possible, then calculating the similarity trajectory matrix of each trajectory through five algorithms, and finally cluster by CLUTO. According to (8), the error rate of each algorithm is obtained. There are 19 clusters of CROSS datasets, and the number of trajectories in each cluster is roughly the same. The data is shown in the **Table 1**.

Table 1. The categories and trajectories of CROSS dataset

Category code	Trajectory number	Category code	Trajectory number
1	103	11	100
2	100	12	100
3	100	13	100
4	100	14	100
5	100	15	97
6	100	16	100
7	100	17	100
8	100	18	100
9	100	19	100
10	100		

LCSS and EDR algorithms need to specify the distance threshold ε, and SDTW and ST-DTW algorithms need to give the weights of sub-distance. In this experiment, the distance threshold ε is between 10^{-6} and 10^{-5}, the weights of sub-distance are between 1 and 100. The five algorithms all need σ, in this experiment, the σ value ranges from 50 to 300 considering the trajectory distance calculated by the four algorithms. The RB and RBR algorithms are used to cluster the trajectories, and multiple experiments are performed to take the result with the smallest error rate as the final result. **Fig. 11**shows the error rate of CROSS dataset clustered by five algorithms. The error rates of the LCSS, EDR, DTW, SDTW, and ST-DTW algorithms in **Fig. 11** are successively decreased, and the DTW, SDTW, and ST-DTW algorithms based on time warping are superior to the other two algorithms. The segment-based algorithms SDTW and ST-DTW are superior to the other point-based algorithms, and the ST-DTW algorithm is superior to the SDTW algorithm. In summary, the ST-DTW algorithm has a better clustering effect in the case of uniform sampling. The specific reason is that the ST-DTW algorithm splits the overall distance into four different dimensions, which represent the direction, length, horizontal spacing, and vertical spacing. By adjusting different weight values, different types of trajectories can be distinguished more accurately.

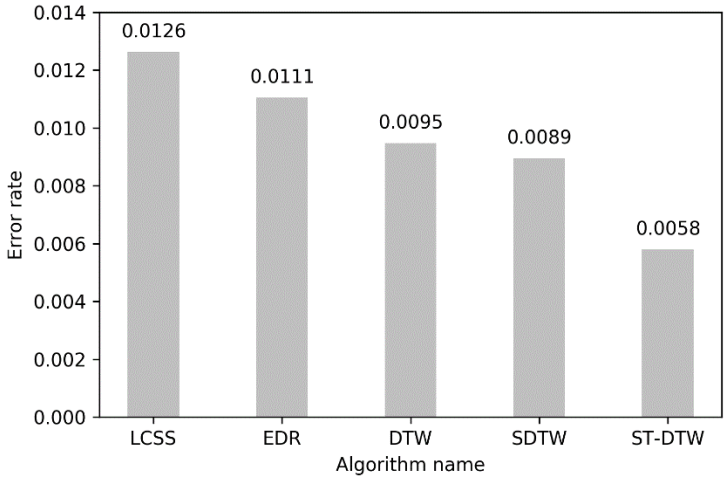

Fig. 11. Error rate of clustering results

5. Conclusions

Trajectory similarity calculation is an important part of trajectory clustering, and trajectory clustering is an inevitable prerequisite for data mining of trajectories, so the study of trajectory similarity is of great significance. The segment transformation distance proposed in this paper can be adapted to different application scenarios by adjusting the weight ratio. The angle distance can be used to distinguish the trajectories in different directions. Scaling distance can be used to distinguish different trajectories of different lengths. The horizontal shift distance can distinguish the trajectory with a large difference between the starting points (end points) of the two trajectory segments and the vertical shift distance can better calculate the trajectories running side by side on the same main road. Using different weights respectively for different characteristics of the trajectory can make the algorithm a better adaptability and accuracy.

In the future, the trajectory similarity calculation methods proposed in this paper can be improved in the following two ways:

(1)Find the optimal parameters. Reasonable parameters can guarantee the experimental results while achieving less time. However, the setting of parameters does not have a clear guiding ideology in this paper. Based on the reference of geometric sense, it is recommended to increase the weight of the angle distance appropriately. Therefore, how to reasonably select parameter values according to the characteristics of the data set itself is the next step to be taken. For example, heuristic algorithms such as genetic algorithms can be used for further testing in the future.

(2)Reduce the error caused by sampling points. Segment-based trajectory clustering has a distinct disadvantage which is when the sampling points differ greatly, the calculation result will have a large deviation. For example, a same trajectory, one with two recorded points, and the other with five recorded points, the trajectory distance between them will have a large error. For this case, the method of interpolating and adding sampling to the trajectory with less recorded points can be solved.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this articles.

References

[1] J. Bao, T. He, S. Ruan, Y. Li, Y. Zheng, "Planning Bike Lanes based on Sharing-Bikes' Trajectories," in *Proc. of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (ACM SIGKDD 2017)*, Halifax, Canada, pp. 1377-1386, 2017. Article (CrossRef Link)

[2] A. Abadi, T. Rajabioun, P. Ioannou, "Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015. Article (CrossRef Link)

[3] J. Zhang, Y. Zheng, D. Qi, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," in *Proc. of 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, San Francisco, California, USA, pp. 1655–1661, 2017. Article (CrossRef Link)

[4] D. Yang, D. Zhang, Z. Yu, Z. Yu, "Fine-Grained preference-aware location search leveraging crowdsourced digital footprints from LBSNs," in *Proc. of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013)*, Zurich, Switzerland, pp. 479-488, 2013. Article (CrossRef Link)

[5] Nicholas J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, H. Xiong, "Discovering Urban Functional Zones Using Latent Activity Trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 712–725, 2015. Article (CrossRef Link)

[6] JD. Chen, "Clustering Objects in a Road Network," *Journal of Software*, vol. 18, no. 2, pp. 332–344, 2007.

[7] R. Agrawal, C. Faloutsos, A. Swami, "Efficient similarity search in sequence databases," in *Proc. of the International Conference on Foundations of Data Organization and Algorithms (FDOA 1993)*, Chicago, Illinois, USA, pp. 69–84, 1993. Article (CrossRef Link)

[8] SL. Lee, SJ. Chun, DH. Kim, JH. Lee, CW. Chung, "Similarity search for multidimensional data sequences," in *Proc. of 16th International Conference on Data Engineering (ICDE 2000)*, San Diego, CA, USA, pp. 599-608, 2000. Article (CrossRef Link)

[9] R. Agrawal, KI. Lin, HS. Sawhney, K. Shim, "Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases," in *Proc. of the 21th International Conference on Very Large Data Bases (VLDB 1995)*, San Francisco, CA, USA, pp. 490-501, 1995. Article (CrossRef Link)

[10] L. Chen, R. Ng, "On the Marriage of Lp-norms and Edit Distance," in *Proc. of the Thirtieth International Conference on Very Large Data Bases (VLDB 2004)*, Toronto, Canada, pp. 792-803, 2004. Article (CrossRef Link)

[11] L. Chen, MT. Özsu, V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proc. of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD 2005)*, Baltimore, Maryland, USA, pp. 491-502, 2005. Article (CrossRef Link)

[12] JG. Lee, J. Han, KY. Whang, "Trajectory clustering: a partition-and-group framework," in *Proc. of the 2007 ACM SIGMOD international conference on management of data (SIGMOD 2007)*, Beijing, China, pp. 593-604, 2007. Article (CrossRef Link)

[13] BT. Morris, MM. Trivedi, "Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and Long-Term Adaptive Approach," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, no. 11, pp. 2287-2301, 2011. Article (CrossRef Link)

[14] X. Song, Y. Pu, D. Liu, Y. Feng, "Mining urban functional areas using pedestrians' movement trajectories," *Acta Geodaetica et Cartographica Sinica*, vol. 44, pp. 82–88, 2015.

[15] M. Ester, HP. Kriegel, J. Sander, X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of the Second International Conference on Knowledge Discovery and Data Mining (ACM KDD)*, Portland, Oregon, pp. 226–231, 1996. Article (CrossRef Link)
[16] J. Gan, Y. Tao, "DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation," in *Proc. of the 2015 ACM SIGMOD international conference on management of data (SIGMOD 2015)*, New York, USA, pp. 519-530, 2015. Article (CrossRef Link)

LONGBAO WANG is currently a senior engineer with the College of Computer and Information, Hohai University. He has published over 40 papers. His research interests include water conservancy big data and domain software.

XIN LV is currently an Associate Professor with the College of Computer and Information, Hohai University. He has published over 60 papers. His research interests include cryptography, network information security, and privacy-preserving theory and technology.

JICUN AN received the B.E. and master's degrees in computer science and technology from Hohai University, Nanjing, China. His research interests include multi-source data fusion and big data analysis.