Research on The Effectiveness of Trajectory Similarity Measurement Algorithm

Yanfu Wu

1College of Transport & Communications, Shanghai Maritime University, Shanghai, 201306, China

*Corresponding author’s e-mail: 202030610134@stu.shmtu.edu.cn

Abstract. Under the background of big data era, massive trajectory data is constantly generated, which contains great value. The analysis and mining of spatiotemporal trajectory data is a research hotspot of spatial data, including trajectory retrieval, trajectory classification etc. In the process of analysis and mining, the similarity measurement between different trajectories is a key problem. Considering the amount of data, computational complexity, noise and other factors, different metrics need to be selected in different situations. In order to understand the performance of different similarity measurement algorithms, this paper selects the Longest Common Sub-Sequence Algorithm (LCSS), Fréchet Distance Algorithm and One Way Distance Algorithm (OWD) among the three kinds of trajectory similarity measurement algorithms and they were used for comparison. Experimental results show that LCSS algorithm can effectively resist the interference of noise points, Fréchet Distance Algorithm has strong robustness, and OWD Algorithm has low time complexity and short execution time.

1. Introduction

In the era of big data, massive trajectory data is constantly generated, which contains great value, such as traffic flow analysis and prediction, providing suggestions for the government's urban planning; it can also cluster tracks to find the roads that are passed by many tracks, which can be used to guide the planning of bicycle lanes; It can also detect the dwell point and find the region where the trajectory often stays. The analysis and processing of trajectory data is very challenging, which mainly includes three aspects: 1) large amount of trajectory data; 2) The track data is noisy; 3) There are many ways to obtain trajectory data. Among them, as a basic algorithm service, trajectory similarity measures the distance between two tracks, which can provide support for its upper application, and is also one of the current research hotspots.

2. Algorithm Introduction

Compared with the distance measurement between points or between points and trajectories, the distance measurement between trajectories is more complex and needs to consider more factors, such as the sampling rate of trajectories, the time information of trajectories and the noise of trajectories. The common measurement methods of trajectory similarity are roughly classified as shown in the figure below:
2.1. Longest common sub-sequence algorithm

There is a classical algorithm problem: to solve the longest common subsequence of two sequences, do not require two consecutive subsequences in the common subsequence, for example, the largest common subsequence of bdcaba and abcbdbab is bcba. On this basis, it is natural to propose a trajectory similarity measurement method based on the longest common subsequence, namely LCSS, whose value represents the number of points that can be regarded as the same point at most, that is, the logarithm of the trajectory points satisfying the minimum distance threshold limit in two trajectories. The algorithm based on dynamic programming is as follows:

\[
d_{\text{LCSS}}(tr_1, tr_2) = \begin{cases} 
0, & \text{if } n = 0 \text{ or } m = 0 \\
1 + d_{\text{LCSS}}(\text{Rest}(tr_1), \text{Rest}(tr_2)), & \text{if } d_{\text{LCSS}}(\text{Head}(tr_1), \text{Head}(tr_2)) \leq \varepsilon \\
\max \left\{ d_{\text{LCSS}}(\text{Rest}(tr_1), \text{Rest}(tr_2)), d_{\text{LCSS}}(\text{tr}_1, \text{Rest}(tr_2)) \right\}, & \text{other}
\end{cases}
\]

The parameter \(\varepsilon\) is the minimum distance threshold, when the distance between two points is less than this value, it will be considered as the same point. In addition, the algorithm has no limit on the length of the trajectory.
Figure 2. Schematic diagram of LCSS algorithm.

2.2. Fréchet Distance Algorithm

Intuitively, Fréchet Distance is the dog rope distance, that is, the owner’s walking path A, the dog’s walking path B, the shortest length of the dog rope needed in the process of completing the two paths.

Figure 3. Schematic diagram of dog rope distance.

The algorithm based on dynamic programming is as follows:

\[
d_F(tr_1, tr_2) = \begin{cases} 
\max_{1 \leq i \leq n} d(p_i^1, p_i^2), m = 1 \\
\max_{1 \leq i \leq m} d(p_i^1, p_i^2), n = 1 \\
\max \left\{ \begin{array}{l}
\min \left\{ \begin{array}{l}
d_F(tr_1^{n-1}, tr_2), \\
d_F(tr_1, tr_2^{m-1}), \\
d_F(tr_1^{n-1}, tr_2^{m-1})
\end{array} \right\} , \text{other}
\end{array} \right\}
\end{cases}
\]

And \(d(p, q)\) is the Euclidean distance between two GPS points, trajectories of length N-1 of trajectories tr is \(tr(n-1) = \langle p_1 \rightarrow p_2 \cdots p_{n-1} \rangle\).

2.3. One way distance algorithm

The basic idea of owd distance is based on the area surrounded by two tracks. When the area is large, it indicates that the distance between tracks is far, and the similarity is low; On the contrary, if the enclosed area is 0, the two tracks coincide and the similarity is the highest. The definition of one-way distance is as follows:
\[
OWD(t_1, t_2) = \frac{1}{|t_1|} \int_{p \in t_1} d(p, t_2) dp
\]

For symmetry, the above formula can be changed as follows:
\[
d_{OWD}(t_1, t_2) = \frac{1}{2}(OWD(t_1, t_2) + OWD(t_2, t_1))
\]

3. Comparative experiment and result analysis

In this section, we set up two groups of experiments, which are the comparison of algorithm execution time and algorithm robustness. The experimental data are two real AIS trajectories. Track \( t_1 \) has 980 trajectories and track \( t_2 \) has 950 trajectories. And 20 parallel control groups were set up in each group to take the mean value.

3.1. Algorithm execution time comparison experiment

In this group of experiments, three algorithms are used to calculate the track similarity of two original tracks, and record the algorithm running time data.

| Algorithm          | Execution time (s) |
|--------------------|--------------------|
| LCSS               | 0.7255             |
| Fréchet Distance   | 0.7860             |
| OWD                | 0.0155             |

Experiments show that the execution time of OWD algorithm is much lower than that of LCSS and Fréchet Distance. In the case of the same amount of data, the execution time of OWD is reduced by nearly 90%.

3.2. Algorithm robustness comparison experiment

In this group of experiments, 0, 300, 400 and 500 trajectory points were randomly deleted from two original trajectories, and the robustness of the algorithm in different degrees of missing trajectory data was observed by comparing the experimental data.

| Number of delete points | Trajectory similarity (%) |
|-------------------------|---------------------------|
|                         | LCSS          | Fréchet Distance | OWD           |
| 0                       | 85.76         | 91.33           | 97.42         |
| 300                     | 81.90         | 90.85           | 97.35         |
| 400                     | 75.19         | 91.05           | 97.33         |
| 500                     | 66.05         | 90.01           | 97.33         |

It can be seen from the experimental data that the similarity measure changes after the missing trajectory data. The results show that the similarity measure of LCSS algorithm after missing trajectory data is significantly reduced, and there is an obvious error in the similarity judgment after missing. And Fréchet Distance algorithm and OWD algorithm have good robustness in the case of random missing trajectory data.

4. Conclusion

In terms of execution time, OWD algorithm is much better than LCSS algorithm and Fréchet Distance algorithm. And in a more realistic situation, the trajectory data will be much larger than the amount of
data involved in the experiment, and the difference in the execution time of these three algorithms will be more huge. In the robustness of the algorithm, OWD algorithm and Fréchet Distance algorithm can maintain good robustness in the trajectory missing experiment, and the measurement of trajectory similarity can be relatively stable to a certain extent. However, when the missing track data increases gradually, the LCSS measurement value of track similarity decreases significantly, which indicates that the robustness of LCSS algorithm is low.

References
[1] On the marriage of LP-norms and edit distance. Chen L, Ng R. Proceedings of the Thirtieth international conference on Very large data bases. 2004
[2] Robust and Fast Similarity Search for Moving Object Trajectories. Lei Chen, M Tamer Ozu, Vincent Oria. Proceedings of the International Conference on Management of Data. 2005
[3] One Way Distance: For Shape Based Similarity Search of Moving Object Trajectories. Bin Lin, Jianwen Su. GeoInformatica. 2008 (2)
[4] Searching Trajectories by Locations: An Efficiency Study. Zaiben Chen, Hengtao Shen, Xiaofang Zhou, Yu Zheng, Xing Xie. Proceedings of the ACM SIGMOD International Conference on Management of Data. 2010
[5] Algorithms for the Longest Common Subsequence Problem. Daniel S. Hirschberg. Journal of the ACM (JACM). 1977 (4)
[6] An effectiveness study on trajectory similarity measures. WANG H, SU H, ZHENG K, et al. Proceedings of the 24th Australasian Database Conference. 2013