Trajectory Clustering Based on Trajectory Structure and Longest Common Subsequence

Guang-ya ZHANG* and Jing ZHANG
Information Engineering University, Zhengzhou, Henan, China
*Corresponding author

Keywords: Trajectory clustering, Density-based clustering, Longest common subsequence.

Abstract. Trajectory clustering is an important method for mining valuable information from spatio-temporal data. Longest common subsequence clustering algorithm has advantages in distinguishing the overall trend of the trajectory, but ignores the trajectory structure details. At the same time, trajectory structural similarity clustering algorithm performs recognizing track structural details better than overall trends. Therefore we come up with a trajectory clustering algorithm based on trajectory structure and longest common subsequence in order to combine the advantages of them. The point distance calculation in LCSS is changed to trajectory structural distance which reflects the trajectory details better. The partition algorithm in trajectory structure clustering is improved to reduce the influence of noise points. The simulation results shows that TS-LCSS performs better than TC-SS when we need to recognize trajectories’ over all trends.

Introduction

Trajectory clustering is the process of extracting the similarity, anomaly and valuable patterns from the trajectory data. It is widely used in research and engineering [1]. There are lots of classical clustering algorithms, like regression mixture model [2], partition-and-group framework and micro-and-macro cluster framework [3-5], while a number of improved algorithms have been proposed in literatures. YUAN Guan et al. divided the trajectory by the angle of rotation, and then proposed structural similarity (SSIM) function for clustering. Utilizing this clustering algorithm, the trajectory’s local similarity could be found but the overall characteristics are ignored [6]. The longest common subsequence (LCSS) algorithm is used to solve the problem that some noise points in the trajectory leads to a great difference in the overall trajectory [7]. It can also be used to distinguish the direction of trajectories. Wei et al. proposed to cluster by trajectory Hausdorff distance first, and then use LCSS clustering to distinguish the trajectory direction and improve the accuracy of trajectory classification [8]. The connection between the trajectory points contains fatal features of the trajectory, but the LCSS algorithm is just based on the Euclidean distance between the points. The method of clustering by location used in this algorithm is also based on the entire trajectory. Therefore, this algorithm ignores the local features of the trajectory.

This paper combines the advantages of the two algorithms above and proposes a trajectory cluster method that based on trajectory structure similarity and LCSS. The Euclidean distance in traditional LCSS measures trajectories by point data, which cannot fully reflect the structure of trajectory. Therefore we use trajectory structure similarity between sub-trajectories instead of it. For the traditional trajectory segmentation algorithm, the interference of the noise points often cause wrong segmentation, so we partition the trajectory by the integral trajectory angle instead of the adjacent trajectory angle. We also improved the two-step clustering method. Firstly, the improved LCSS trajectory clustering is used to obtain the rough clustering. Then we performed trajectory segmentation and clustered the trajectory segments based on the trajectory structure similarity. The algorithm combines the advantages of LCSS algorithm in identifying the trajectories’ overall characteristics and dealing with unsimilar parts in the trajectory, which improves the clustering accuracy.
Modified Longest Common Subsequence Algorithm

LCSS is, just as the name indicates, the longest common subsequence of two ordered sets. It is used in spatio-temporal data clustering to solve the problem that some noise points in the trajectory leads to a great difference in the overall trajectory. So we use LCSS to group trajectories by the overall trends.

![Figure 1. Diagram of LCSS.](image)

The main idea of LCSS is to calculate the Euclidean distance of the points in two trajectories in turn. When the distance is under a threshold, we find one part of common subsequence. At last we calculate the length of longest common subsequence. Figure 1 shows the algorithm of LCSS.

LCSS is usually calculated in recursion way as Eq.1.

\[
\text{LCSS}(\text{Res}(L_i)[N], \text{Res}(L_j)[M]) = \begin{cases} 
0, M = 0 \text{ or } N = 0 \\
\text{LCSS}(\text{Res}(L_i)[N-1], \text{Res}(L_j)[M-1]) + 1, \text{dist}(p_a, p_b) < \varepsilon \\
\text{max}\left\{\text{LCSS}(\text{Res}(L_i)[N-1], \text{Res}(L_j)[M]), \text{LCSS}(\text{Res}(L_i)[N], \text{Res}(L_j)[M-1])\right\}, \text{else}
\end{cases}
\]

(1)

In the equation \(\text{Res}(L_i)[N]\) denotes the \(N^{th}\) point of trajectory \(L_i\), or \(p_a\) in short. \(\varepsilon\) denotes the threshold of similarity of distance. And \(\text{dist}(p_a, p_b)\) stands for the Euclidean distance between point \(p_a\) and \(p_b\).

However, this algorithm uses Euclidean distance between points to calculate LCSS, which ignores the critical value of trajectory structure. Therefore, in order to optimize the LCSS algorithm we deeply integrates it with the trajectory structure computation. The distance between points is changed into the distance metric between two points, and the general Euclidean distance is changed to the trajectory structural distance measure.

In this way, Eq. 1 is transformed into Eq.2.

\[
\text{LCSS}(\text{He}(L_i)[x], \text{He}(L_j)[y]) = \begin{cases} 
0, x = 0 \text{ or } y = 0 \\
\text{LCSS}(\text{He}(L_i)[x-1], \text{He}(L_j)[y-1]) + 1, TS(\text{Res}(L_i)[x], \text{Res}(L_j)[y]) < \sigma \\
\text{max}\left\{\text{LCSS}(\text{He}(L_i)[x-1], \text{He}(L_j)[y]), \text{LCSS}(\text{He}(L_i)[x], \text{He}(L_j)[y-1])\right\}, \text{else}
\end{cases}
\]

(2)

\(TS\) is the trajectory structural distance proposed by Lee et al. which inculdes three parts.
Modified Trajectory Partition Algorithm

The purpose of trajectory segmentation is to divide the whole trajectory into sub-trajectories in which targets move in the similar motion feature. So the key to the segmentation is to find the point where the target’s motion feature changes rapidly. The angle between two track segments can reflect the target’s movement trend. So YUAN et al. come up with a trajectory partition method based on the angle, like angle $\alpha_2$ showed in Figure 3. For the sub-trajectory begin with $p_{cj}$, $\alpha_2$ equals to the angle $\angle(p_j, p_{j+1})$. The interference of noise may leads to wrong partition, so we improve this algorithm. For point $p_j$, the angle between $p_{cj}p_j$ and $p_jp_{j+1}$, denoted as $\alpha_1$, is calculated instead of $\alpha_2$. In this way, the noise points influence less on the trajectory trend, while the computational complexity keeps on the level of $O(n)$. At the same time, the threshold should be reduced compared with traditional algorithms.

![Figure 3. Diagram of Trajectory Partition.](image)

Trajectory Clustering Algorithm Based on Trajectory Structure and LCSS

There are two steps in the trajectory clustering algorithm based on trajectory structure and LCSS (TS-LCSS). Primarily, modified LCSS is used to cluster the trajectories roughly by their overall trends. Then the trajectories are partitioned and the sub-trajectories in each rough clusters are clustered by the trajectory structural similarity.

The structural similarity is defined as below in Eq.7. The characteristic distance (CDIST) is used in the definition.

$$CDIST(L_i, L_j) = (DirDist + SpeedDist + AngleDist + LocDist) \times W^T$$

$$LSIM(L_i, L_j) = 1 - Normalized(CDIST(L_i, L_j))$$

In Eq.7, $DirDist$ denotes the directional distance, $SpeedDist$ denotes the distance of speed, $AngleDist$ denotes the distance of angle, and $LocDist$ denotes the distance of location.
$W=[W_D, W_S, W_A, W_L]$ is the weighted vector to adjust the importance of different parameters. $\text{Normalized}(CDIST(L_i, L_j))$ is the normalization function to balance the parameters' value.

The complete algorithm is listed below.

**Algorithm TS-LCSS**

**INPUT:** Trajectories sets $D = \{tr_1, tr_2, ..., tr_N\}$. Each trajectory is denoted as $tr_i = p_1 p_2 ... p_n$

**OUTPUT:** A set of clusters $C = \{c_1, c_2, ..., c_N\}$. Each cluster contains several trajectories $c_j = \{tr_a, tr_b, ...\}$

**ALGORITHM:**

1. Calculate the modified LCSS distance between trajectories to get the distance matrix $\text{DIST}_{L} = [\text{dist}_{L}]_{N \times N}$.
2. Cluster the trajectories with $\text{DIST}_{L}$ using density based clustering algorithm and get the rough cluster set $C = \{c_{i1}, c_{i2}, ..., c_{ik}, ...\}$.
3. Partition the trajectories into sub-trajectories using modified partition algorithm.
4. Calculate the structural matrix $\text{LSIM}(tk) = [\text{LSIM}(tr_a, tr_b)]$ in every rough cluster $c_{ik}$.
5. Cluster the sub-trajectories using density based clustering algorithm using $\text{LSIM}(tk)$ and output the cluster set $C$.

**Simulations**

We use simulated data sets to show the correct clustering rate (CCR) of different algorithm. And 60 trajectories in the hurricane dataset is used to compare the clustering results of different algorithm. The correct clustering rate is defined in Equ.9. TCC stands for total correct clustering. TSN stands for total samples number and ACP denotes the average coordinates points.

$$\text{CCR} = \frac{TCC}{TSN} \quad (9)$$

![Figure 4. Comparison of CCR.](image)

Figure 4 reveal the performance comparison between TS-LCSS and TS-SS. It shows that TS-LCSS has a higher correct clustering rate than TC-SS when the ACP raises up.
Figure 5 reveals that TS-LCSS has the ability to cluster more trajectories and find the trajectories with the same overall trends but in different length.

**Summary**

In this paper we come up with an algorithm which combines the advantages of trajectory structural similarity and longest common subsequence. The LCSS is modified to reflect the trajectory better and the partition algorithm is improved to reduce the influence of noise points. The simulation results shows that TS-LCSS performs better than TC-SS when we need to recognize different directions of trajectories.

**Acknowledgement**

This research was financially supported by the National Science Foundation (61601516).

**References**

[1] Han, J., Kamber, M. Data mining: concepts and techniques. J. Data Mining Concepts Models Methods & Algorithms Second Edition, 5(4) (2011) 1-18.

[2] Gaffney, S. J. Probabilistic Curve-Aligned Clustering and Prediction with Regression Mixture Models. Ph.D. Dissertation. Laboratoire MAS. 8 (2004) 391.

[3] Lee, J. G., Han, J., Whang, K. Y. Trajectory clustering: a partition-and-group framework. ACM SIGMOD International Conference on Management of Data (2007) 593-604.

[4] Li, Z., JaeGil Lee, Li, X., Han, J. Incremental Clustering for Trajectories. International Conference on Database Systems for Advanced Applications. Springer-Verlag 5982 (2010) 32-46.

[5] Yuan, G., Sun, P., Zhao, J., Li, D., Wang, C. A review of moving object trajectory clustering algorithms. Artificial Intelligence Review, 47(1) (2017) 123-144.

[6] Guan, Y., Xia, S. X., Lei, Z., Yong, Z. Trajectory clustering algorithm based on structural similarity. Journal on Communications. 32(9) (2011) 103-110.

[7] Gudmundsson, J. Computing longest duration flocks in trajectory data. ACM International Symposium on Advances in Geographic Information Systems. (2006) 35-42.

[8] Wei, L. X., He, X. H., Teng, Q. Z., Gao, M. L. Trajectory classification based on hausdorff distance and longest common subsequence. Journal of Electronics & Information Technology. 35(4) (2013) 784-790.