


Article

A Trajectory Regression Clustering Technique Combining a Novel Fuzzy C-Means Clustering Algorithm with the Least Squares Method

Xiangbing Zhou ^{1,2,3,*} , Fang Miao ², Hongjiang Ma ⁴, Hua Zhang ¹ and Huaming Gong ³

¹ School of Information and Engineering, Sichuan Tourism University, Chengdu 610100, China; zhangh@abtc.edu.cn

² Key Lab of Earth Exploration & Information Techniques of Ministry Education, Chengdu University of Technology, Chengdu 610059, China; miaof@abtc.edu.cn

³ School of Mathematics and Computer Science, Aba Teachers University, Wenchuan 623002, China; gonghm@abtc.edu.cn

⁴ School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China; mahj@abtc.edu.cn

* Correspondence: zhoubx@abtc.edu.cn; Tel.: +86-186-028-00090

Received: 11 March 2018; Accepted: 23 April 2018; Published: 25 April 2018



Abstract: Rapidly growing GPS (Global Positioning System) trajectories hide much valuable information, such as city road planning, urban travel demand, and population migration. In order to mine the hidden information and to capture better clustering results, a trajectory regression clustering method (an unsupervised trajectory clustering method) is proposed to reduce local information loss of the trajectory and to avoid getting stuck in the local optimum. Using this method, we first define our new concept of trajectory clustering and construct a novel partitioning (angle-based partitioning) method of line segments; second, the Lagrange-based method and Hausdorff-based K-means++ are integrated in fuzzy C-means (FCM) clustering, which are used to maintain the stability and the robustness of the clustering process; finally, least squares regression model is employed to achieve regression clustering of the trajectory. In our experiment, the performance and effectiveness of our method is validated against real-world taxi GPS data. When comparing our clustering algorithm with the partition-based clustering algorithms (K-means, K-median, and FCM), our experimental results demonstrate that the presented method is more effective and generates a more reasonable trajectory.

Keywords: trajectory regression clustering; Hausdorff distance; angle-based line segments partitioning; Lagrange-based fuzzy C-means; least squares regression; taxi GPS data

1. Introduction

In recent years, the increasing popularity of GPS (Global Position System)-enabled devices has facilitated users to track moving objects on the internet. Typically, GPS-device taxis are widely used in many cities [1], and record GPS information and movement trajectories that could reflect city states, such as traffic congestion [2–4], urban travel demand and transport services [5,6], and population migration distribution [7,8]. Issues concerning how to mine the hidden information and understand the meaning of these states, as well as how the information of trajectories can be employed in urban development have become research hotspots. Therefore, a great quantity of clustering-based approaches was presented to describe states of the city, which utilized characteristics and trajectory pattern clustering of GPS data [1,6,9–16]. For example, Reference [10] presented a density-based line segments trajectories clustering algorithm that was based on a partition and group framework. The authors in Reference [9] presented a two-step clustering algorithm that was based on density,

which was made of segment and trajectory clustering. The authors in Reference [12] presented a road network aware approach for the fast and effective clustering of road segment spatial trajectories, which was used to replace density-based clustering and Euclidean-based distance computing. Reference [11] presented a scalable and fast density clustering algorithm that was based on big data computing. Also, the authors of Reference [16] presented an improved density-based algorithm that was to be used for stops clustering in trajectories. In particular, work in Reference [17] proposed an anisotropic (angle-based standard deviation) density-based clustering algorithm, which was used to discover spatial point patterns with noise.

In general, there are several categories of clustering algorithms [17,18]; density-based, partitioning-based, grid-based, hierarchical-based, and graph-based, which have a wide range of applications in spatial data processing [19,20]. Furthermore, each of these categories contains several well-known clustering algorithms (e.g., partitioning-based K-means, K-median, and fuzzy C-means (FCM)), with their specific pros and cons. In particular, density-based clustering algorithms are usually used to mine the hidden information of a given dataset and handle any GPS datasets, as they are particularly suitable for discovering clusters with arbitrary shapes and finding mutual exclusion clusters [9–11,16,17]. However, it is difficult to handle the overlapping clusters (e.g., trajectory crossover), when considering fuzzy clusters and the loss of local information of trajectories. In addition, it is sensitive to the set neighborhood and the density (*MinPts*). In this paper, we focus on the partitioning-based approaches (e.g., FCM). However, they still have several shortcomings, including sensitivity to the initial cluster centers selection, slowness of convergence, and a tendency to become stuck in the local optimum. Therefore, In this paper, a novel trajectory regression clustering technique that is based on partition clustering is proposed, which combines a new line segments (based on angle) produced method (AngPart), a Lagrange-based fuzzy C-means clustering (FCML) algorithm, and the least squares regression model (LSR), which is used to construct an unsupervised trajectory clustering method instead of the map-based knowledge base. Namely, FCML is a novel unsupervised partitioning regression clustering algorithm that combines AngPart and FCM with LSR, which is shown in Figure 1. Firstly, a line segment partitioning method is constructed, which is used to efficiently produce line segments with three GPS data points (see Section 4.1), and is used to preserve the local information of trajectories. Secondly, the presented novel clustering algorithm combined a novel fuzzy C-means (NFCM) with the Lagrange operator [21] and Hausdorff-based K-means++ [22,23], which is used to capture the global optimum and to avoid getting stuck in local optimum, respectively, where the NFCM is used to achieve line segments clustering and K-means++ are used to produce the initial cluster centers of the line segments. In particular, the original fuzzy C-means (FCM) algorithm is a partitioning-based clustering method [24], the Hausdorff [25] distances computing between line segments must be used to replace the Euclidean distance. Finally, when the hidden information of GPS data is mined and obtained, the LSR is employed to achieve trajectory regression, with its aim to regress and generate trajectories of the clustering results without map-based knowledge base, which can be used to explain and describe urban states (e.g., people, vehicles, roads, traffic flow, and reference as roads planning) around the produced trajectories.

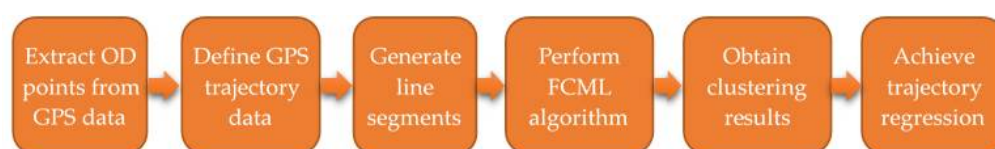


Figure 1. Flowchart of the Lagrange-based fuzzy C-means clustering algorithm (FCML).

In fact, FCML can improve line segment partitioning and preserve the local information of trajectory using the angle-based method before the clustering operation. For example, if two GPS data points are generated as line segments, it is difficult to explain the local information among GPS data

points and to capture the relationship of the successive GPS points (e.g., steering and intersection angle change of the successive GPS points).

In addition, the presented method (FCML) is an unsupervised learning technique. Therefore, when a map-based knowledge base is unnecessary, the least squares regression model (LSR) is used to produce the trajectories of the clustering results.

To verify the performance and effectiveness of FCML, a real-world GPS dataset in Beijing, China, is used as an experimental test (see Section 2), and the experiments as compared FCML with K-median, K-means, and FCM clustering methods using the *PBM* (Pakhira-Bandyopadhyay-Maulik)-*index* [26] cluster evaluation criteria. While *PBM-index* is a very good unsupervised evaluation technique [27,28]; note that distances in the *PBM-index* require the use of the Hausdorff method to calculate between the line segment center and line segments. Also, LSR is used to achieve the regression of the clustering results. The experimental results indicate that FCML achieves better quality trajectory regression than K-means, K-median, and FCM algorithms (see Section 5).

Therefore, the main works of the paper are summarized, as follows:

- (1) A novel line segments generation technique is proposed using the angle-based partitioning method (AngPart).
- (2) A novel fuzzy C-means (NFCM) clustering algorithm is put forth, combining the Lagrange operator with AngPart and K-means++.
- (3) A trajectory regression technique that is based on LSR is presented, which can be used to explain state of population migration around trajectories and can be used as reference for road planning of the city.
- (4) FCML is shown to work on real-world taxi GPS data in Beijing, China.

The rest of the paper is organized as follows. Section 2 describes of taxi GPS data in Beijing, China. Section 3 introduces the angle-based normalizing method that is used for the taxi GPS data. Section 4 proposes a trajectory regression technique combining FCML with LSR. Section 5 presents the experiments and the results for the preformation evaluation of the proposed approaches. Finally, Section 6 concludes the paper and suggests further work.

2. Description of Real-World Taxi GPS Data

The trajectory dataset that was used in this paper was collected from taxi GPS data in Beijing, China [29], the data of which were recorded by different GPS loggers (latitude, longitude) and angles in a given region. The sampling frequency was controlled in two minutes (≤ 2 min); namely, if different sample rates were less than or equal to two minutes, then different location information was recorded, which consisted of the GPS data points of the approximately 30 thousand taxis in 8:50–8:59 a.m. on 20 March 2016. When the origins and destinations (OD) were extracted and mined using a clustering algorithm in Reference [30], this dataset only contained 71,375 OD points in total, as shown in Figure 2. In particular, the OD points are usually used to describe trajectory patterns [8,31]. When the method (angle-based partitioning) in Section 4.1 is performed, the dataset contains 23,785 line segments in total, as shown in Figure 3.

Figure 2 shows the distributions of taxis' OD in a road structure of the given land areas (0.18×0.3) within two minutes in Beijing, China. The overall distributions of OD reflect the traffic change demand of citizens and population migration that use taxicabs as a transportation tool. As a result, traffic information and the population migration distribution can explain the city's situation. When a new road is planned or an old road is improved, it is necessary to consider the traffic status and population migration, with the aim of providing convenient travel and easing traffic congestion. Therefore, in this paper, we present a trajectory regression method that combines FCML with LSR.

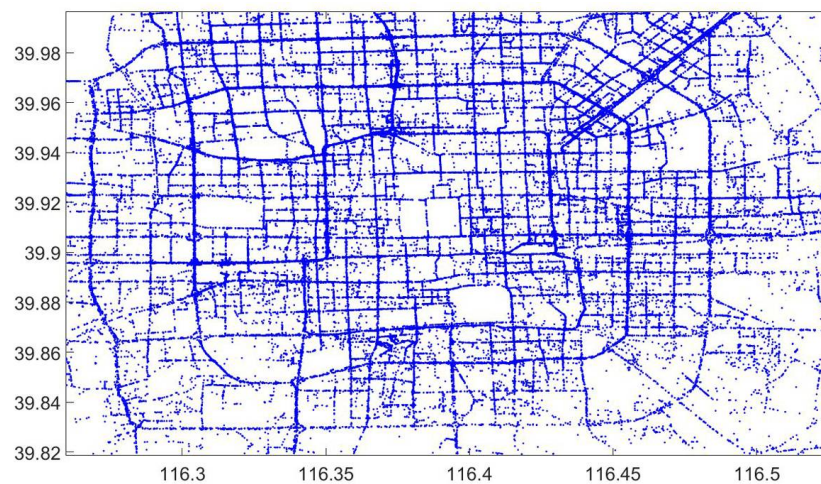


Figure 2. Road structure of the origins and destinations (OD) in Beijing using taxis' GPS (Global Positioning System) data.

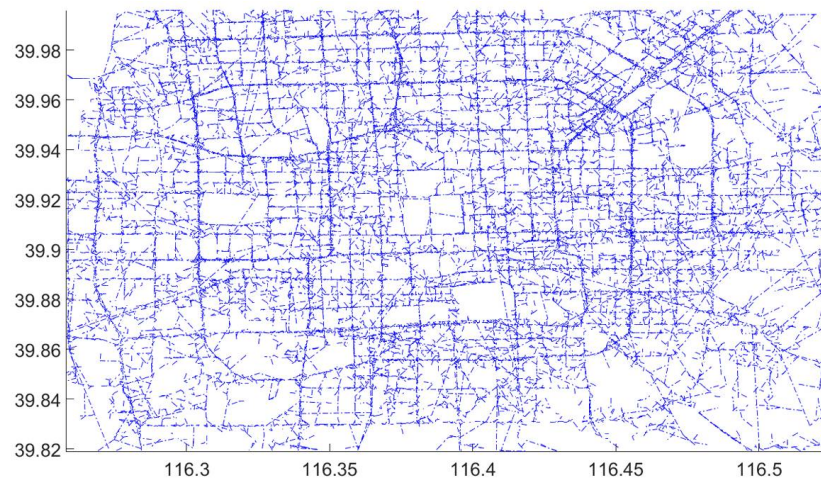


Figure 3. Line segments are produced in term of taxi GPS data points (illustration in Figure 2) using the angle-based approach.

3. Preliminary

In this section, we present the new concepts and operations of the trajectory regression clustering that was used in our technique.

Trajectory: A trajectory is the user-defined GPS point of the evolution of the position of an object that is moving in location during a given time interval in order to achieve a given goal or solve a problem in a geographic information application, e.g., a trajectory can be defined as: $T_i = \{(p_1, t_1), (p_2, t_2), \dots, (p_i, t_i)\}$, where p denotes a pair of GPS points (latitudes, longitudes) and t is the corresponding GPS time [11,20,32]. Or, is described as $T_i = p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_i$ ($p_1 p_2 \dots p_i$), which is a sequence of GPS points in a given time interval [1,10]. In this paper, a taxi trajectory is defined as $T_i = \{(p, a)\} = \{(p_1, a_1), (p_2, a_2), \dots, (p_i, a_i)\}$, or can also be described as $p_1 p_2 \dots p_i : a_1 a_2 \dots a_i \rightarrow p : a$, which represents a sequence of GPS points, where $p = (lng, lat)$ is a GPS point and represents a longitude, latitude location, and a denotes the angle (steering angle) of each taxi GPS point, as illustrated in Figure 4. In addition, we focus on low sampling rate taxi GPS trajectories with $\Delta t \leq 2$ min in order to meet the regression test demand.

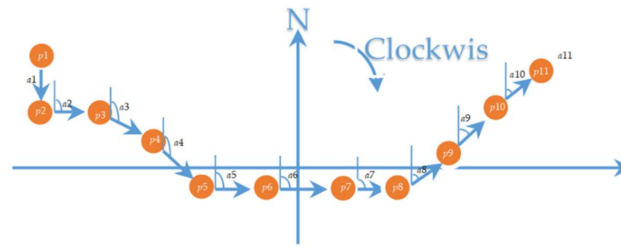


Figure 4. Illustration of the GPS trajectory description based on angle.

Sub-trajectory: A sub-trajectory ST_j is a subset of a trajectory T_i . In this paper, three GPS points that are based on the shortest candidate Euclidean distance constitute a line segment $L_j = \{(p_{j-1}, a_{j-1}), (p_j, a_j), (p_{j+1}, a_{j+1})\} = \{p_{j-1}p_jp_{j+1} : a_{j-1}a_ja_{j+1}\}$, which can also be described as $T_i = \{L_1, L_2, \dots, L_j\} (j \leq i/3)$. Therefore, a combination of $\forall L_{1,\dots,s} (i/3 \geq s \geq 2)$ is considered as a sub-trajectory ST_j of a trajectory T_i . Note that we still use the Euclidean method to calculate the distance between the GPS data points.

L-Similarity: In fact, there exist several methods to measure the similarity of line segments, such as those that are reported in References [13,33,34], as well as cosine similarity. Similar results between line segments are usually used to achieve trajectory clustering and to produce sub-trajectories. In this paper, the given L_j is relevant to angle changes (see Section 4.1). Therefore, the similarity method (similarity measure based on multiple information sources: SMIS) [35] is employed to measure the similarity of line segments, as shown in Equation (1):

$$Sim(L_j', L_j) = \begin{cases} e^{\alpha l} \times \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & L_j' \neq L_j \\ 1 & \text{other} \end{cases} \quad (1)$$

where $\alpha \geq 0$ is a constant and $\beta \geq 0$ is a smoothing factor. $l = \text{siml}(L_j', L_j)$ gives the cosine similarity values between L_j' and L_j ; $h = \min(\text{dist}(L_j', L_j))$ is the minimum Hausdorff distance of GPS points between L_j' and L_j .

Trajectory clustering: A cluster is a set of trajectory partitions. A trajectory partition is a line segment L_j , and the line segments that belong to the same cluster are close to each other in terms of the Hausdorff [25] distance measurement. According to Reference [10], a trajectory can belong to multiple clusters since a trajectory is partitioned into multiple L_j , and trajectory clustering is performed over L_j . A clustering result that is based on line segments $L_{1,2,\dots,j}$ can indicate a common sub-trajectory. Therefore, for given a set of trajectories data (GPS data points) T_i , that are partitioned into many line segments L_j , then a set of clusters can be defined as $C = \{C_1, C_2, \dots, C_K | C_k \subseteq T_i \text{ and } C_k = \{L_{1,2,\dots,k} | k \leq j\}\}$ and the cluster center of each cluster is defined as $c = \{c_1, c_2, \dots, c_K\}$. Namely, a clustering result of line segments is a sequence of GPS points, just like an ordinary trajectory. In particular, in this paper, a cluster center segment is also considered as a line segment without applying a density-based method. An example of trajectory clustering is shown in Figure 5.

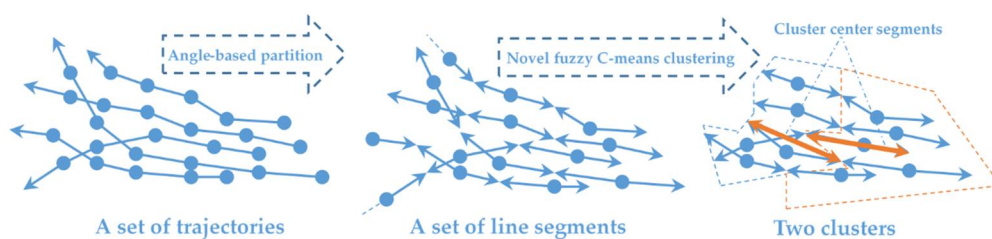


Figure 5. Illustration of trajectory clustering.

4. Methodology

Our methodology is described, as follows: (1) the angle-based partitioning and cosine-based constraint methods are used to generate line segments; (2) Hausdorff-based K-means++ is used to produce initial cluster centers, and a Lagrange-based method is presented to improve FCM clustering; and, (3) the least squares regression method is employed to achieve trajectory regression clustering, as shown in Figure 6.

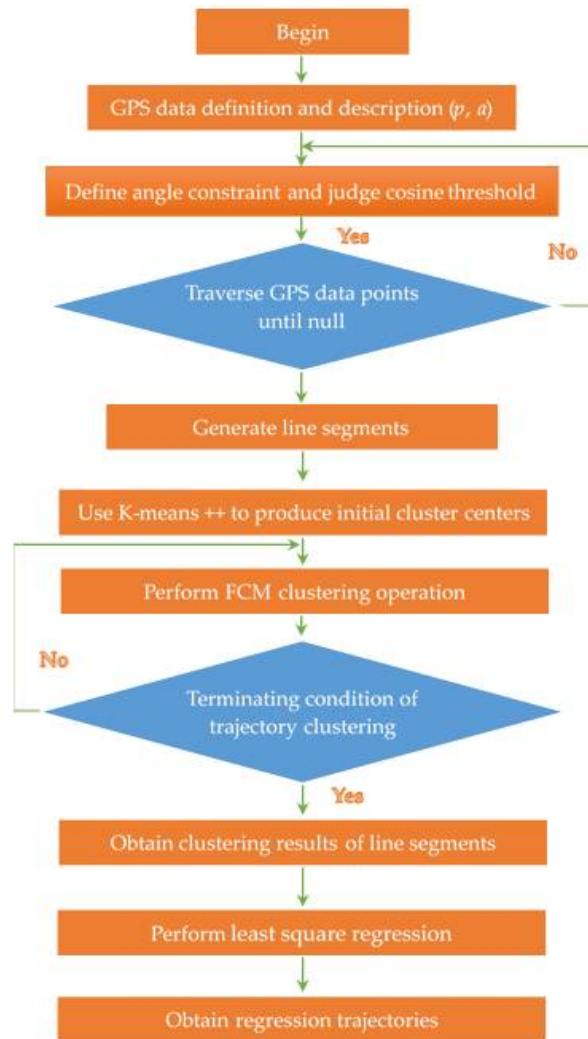


Figure 6. The overall flowchart of our methodology.

4.1. Angle-Based Partitioning and Cosine-Based Constraint

In this section, we present a steering angle-based method used in partitioning line segments, which is composed of two components: (i) $\theta \geq \pi$ and (ii) $\theta < \pi$, where θ denotes the steering angle. Meanwhile, we define a cosine-based method, which is used to restrict the intersection angle γ of three GPS points when the angle threshold is given. The algorithm is shown in Algorithm 1.

Algorithm 1. Angle-based partitioning and cosine-based constraint algorithm

Input: a given GPS dataset D including location information and angles, the number of iterations, the angle threshold T

Output: regression trajectories

Procedure:

```

    Divide into Taxi GPS data: location information and angles, which are set to numbers for each location and angle
    Define list  $Lx$  which is used as temporary storage for three taxi GPS data points
    Define list LineSegment // which is used to store line segments
/* Steering angle-based partitioning */
    WHILE  $D! = \emptyset$  DO
         $P_i$  : count  $\rightarrow D$  // select a data point  $P_i$  from  $D$ , and record the angle of the  $P_i$ ; mark as count =  $i$ ;
        // The "count" is used to count the number of selected data points;
        Distance  $\rightarrow$  Euclidean( $P_i, P_{i+1}$ ) // Calculate distances between the selected  $P_i$  and  $P_{i+1}$ ;
        // select second point  $P_{i+1}$  from  $D$ ;
        // Call the built-in function pdist2 of the Matlab to calculate the Euclidean;
        Sort  $\rightarrow$  Descending(Distance) // Sort the distances in descending order according to the Euclidean;
        DO Angle $i$  : count! = null // Indicate the angle is effective in  $D$ ,
        Calculate : Angle  $\rightarrow \theta$  // Calculate angle difference of the selected data point  $P_i$ ;
        IF  $\theta \geq \pi$ 
            Use Equation (2) to normalize the angles;
        ELSE
            Use Equation (3) to normalize the angles;
        END IF
        WHILE count  $\leq 10$ 
            // the "10" is a given condition, which is used to handle taxi GPS selection
            // we select another 10 GPS data points around first points
            IF Sita  $\leq T$ 
                ( $P_i, P_{i+1}$ )  $\rightarrow Lx$  // Indicates the two taxi GPS data points have been chosen
                // Denotes the shortest distance between  $P_i$  and  $P_{i+1}$ 
            ELSE
                count  $\leftarrow$  count + 1
            END IF
            // when the DO ... WHILE does not satisfy any given values (e.g., 10), then continue to loop
            // is shown in Figure 6
        END WHILE
        IF Judge( $Lx$ ) $i$  == 2 // whether two data points are selected from  $D$ ;
            Select third GPS data point  $P_{i-1}$  as above operation steps
            // above operation steps stand for method of the selected first and second point;
        END IF
        ( $P_{i-1}, P_i, P_{i+1}$ )  $\rightarrow Lx$  // three GPS data points are selected from  $D$ 
        IF Length( $Lx$ ) == 3
             $\gamma \rightarrow$  Cosine( $P_{i-1}, P_i, P_{i+1}$ )
            // Calculate intersection angle between  $P_{i-1}, P_i$  and  $P_i, P_{i+1}$ , where  $P_i$  is a vertex
            IF  $\gamma \geq T$ 
                LineSegment  $\leftarrow (D - Lx(P_{i-1}, P_i, P_{i+1}))$ 
                // Denotes line segment ( $P_{i-1}, P_i, P_{i+1}$ ) is separated from  $D$ , and is stored in LineSegment
            ELSE
                 $D \leftarrow Lx$  // Put  $Lx$  in  $D$ , which are used to recalculate line segments
            END IF
        END IF
         $D \leftarrow Lx$  // Put  $Lx$  in  $D$ 
    END IF
END WHILE
release  $Lx$ 

```


Steering angle-based partitioning: In general, the angles of taxi GPS data points are recorded and collected in terms of the north direction. First, we calculated the shortest distance between the selected first point and another ten candidate GPS data points around the first point (is shown in Figure 7), where “ten” is a given condition (which can also be numbers) that is used to capture angles among the first data point, and then selected the shortest distances between the first point and the other ten candidate data points that are used to handle angle-based partitioning. Therefore, for three selected taxi GPS points, if the bigger angle θ between points is greater than π , it indicates a steering angle in the counter-clockwise direction; if the bigger angle θ between points is less than π , then it indicates a steering angle in the clockwise direction. Then, we need to change the angle using formulas (Equations (2) and (3)), as follows, with the purpose of normalizing the angles and achieving a uniform standard. Illustrations are shown in Figures 8 and 9.

$$Sita = 2\pi - (Sita_2 - Sita_1) \quad (2)$$

$$Sita = Sita_2 - Sita_1 \quad (3)$$

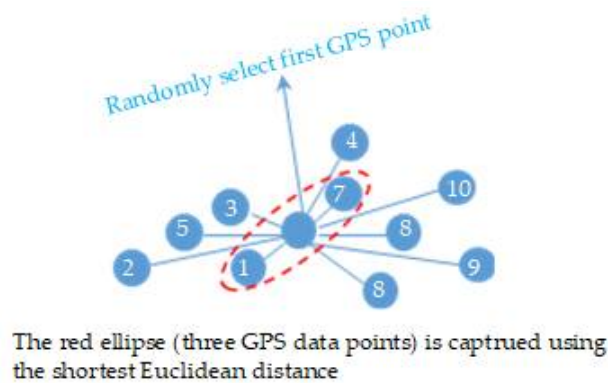


Figure 7. Three GPS data points selection method.

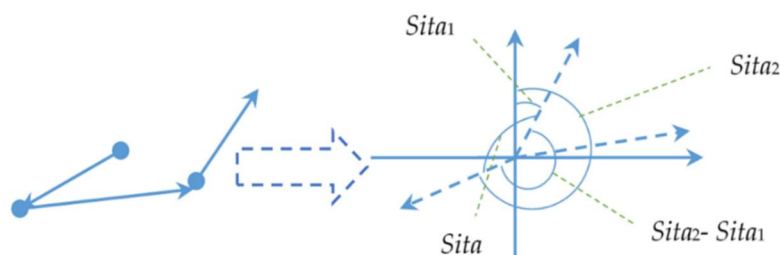


Figure 8. Illustration of (i).

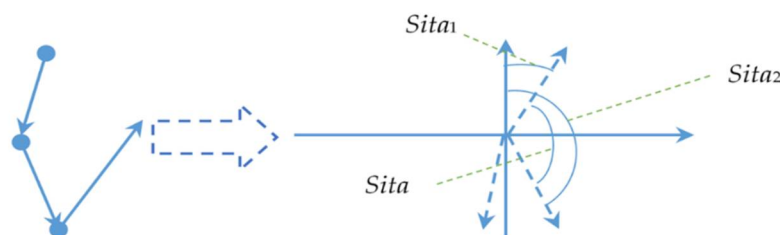


Figure 9. Illustration of (ii).

Intersection angle-based constraint: When the steering angle-based partitions are achieved, the intersection angles $\gamma_t (t = 1, 2, \dots, r)$ needed to be restricted, which are used to explain the movement tendency of the trajectory, as shown in Figure 10. If three taxi GPS points (P^-, P, P^+) are chosen, and P is considered as a vertex, then the intersection angles are defined, as follows (Equation (4)), according to the cosine theorem. First, P is randomly selected from L_x in the Algorithm 1. Second, P is established as a center and then two points are chosen around P , which are captured in terms of the shortest distance that is based on the Euclidean. Third, P^- and P^+ are selected. If $\gamma \leq T$ where T is a given angle threshold (e.g., $T = \frac{\pi}{6}$), then (P^-, P, P^+) is stored in the memory and is separated from the taxi GPS dataset, and then the fourth step is executed; otherwise, we return to the first step. In the fourth step, GPS data points are traversed until the whole dataset is null, which indicates that line segments have been produced.

$$\gamma = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \quad (4)$$

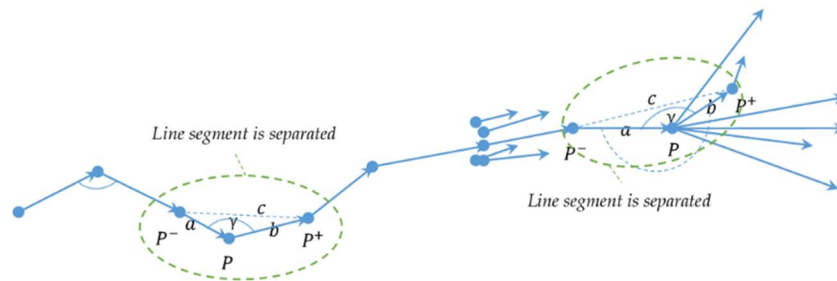


Figure 10. Illustration of intersection angle-based constraint.

As shown in Figure 3, we partitioned the urban area of Beijing into road regions using methods on taxi GPS data points. In other words, the taxi GPS data points of Beijing were divided into line segments using angle-based partitioning and cosine-based constraint methods. When comparing Figure 3 with Figure 2, we found that the local features in Figure 2 were not lost or only a little lost. This indicates that the angle-based method in this paper is effective and feasible (see Section 5).

4.2. Fuzzy C-Means Measure Based on the Lagrange Equation

In general, the clustering step effectively corresponds to the grouping phase and aims to derive a partitioning that is as relevant as possible. However, when the basic partitioning algorithms are used to perform clustering, it is easy to get stuck to the local optima and to suffer from iterative hill-climbing [36–38]. Therefore, we present a novel fuzzy C-means (FCML) clustering algorithm using the Lagrange method, which tries hard to repair the error rates of clustering processing, improve the global optimization, and to balance the iterative hill-climbing. FCM is a partitioning algorithm that allows each data point to belong to multiple clusters with varying degrees of membership [39,40]. In this paper, we seek to improve the FCM in order to achieve line segments clustering and avoid getting stuck in the local optimum, which involved dividing the data points into groups with the most similarities between line segments on the same cluster, and minimum similarities between different clusters, as shown in Equation (5).

$$J_m = \sum_{j=1}^L \sum_{k=1}^K \mu_{jk}^m \|L_j - c_k\|^2 \quad (5)$$

where

(1) $\|\cdot\|$ is any norm expressing the similarity between any measured data and the center.

- (2) L is the number of line segments, which is defined in Section 3.
- (3) K is the number of clusters of line segments.
- (4) m is the fuzzy partition matrix exponent used to control the degree of fuzzy overlap, which in this paper is set to $m = 2$ according to Reference [41].
- (5) L_j and c_k are defined in Section 3.
- (6) μ_{jk} is the degree of membership of L_j in the k th cluster, as shown in Equation (6)

$$\mu_{jk} = \frac{1}{\sum_{k'=1}^K \left(\frac{\|L_j - c_k\|}{\|L_j - c_{k'}\|} \right)^{\frac{2}{m-1}}} \quad (6)$$

To update cluster centers, we combined K-means++ with Equation (7), with the aim of achieving cluster centers to initialize and calculate.

$$c_k = \frac{\sum_{j=1}^L \mu_{ij}^m L_j}{\sum_{j=1}^L \mu_{ij}^m} \quad (7)$$

Then, when K-means++ determines the next line segment, the probability P_+ of K-means++ is given in Equation (8). In addition, we employed Hausdorff to calculate the distance between line segments in K-means++.

$$P'_+ = P_+ \cdot c_k \quad (8)$$

where P'_+ is a fuzzy method used in the next line segment center selection, and $P_+ = \frac{\min(\text{Hausd}(L_{1k}, L_{jk}))}{\sum(\text{Hausd}(L_{1k}, L_{jk}))}$, according to References [22,23]. Finally, Hausd denotes the Hausdorff distance.

To adjust the objective function J_m , the Lagrange operator is presented in Equations (8) and (9).

$$J(g) = J(g) - \rho \cdot \frac{g_{\text{current}}}{g_{\text{max}}} \cdot \sum_{l \in \{1, 2, \dots, K\}} \left(\frac{\Delta J_k}{J_k^{\text{max}}(g) - J_k^{\text{min}}(g)} \right)^2 \quad (0 < \rho < 1) \quad (9)$$

$$\Delta f_k = \begin{cases} J_k - J_k^{\text{max}} & \text{if } J_k > J_k^{\text{max}} \\ 0 & \text{if } J_k^{\text{min}} < J_k < J_k^{\text{max}} \\ J_k^{\text{min}} - J_k & \text{if } J_k < J_k^{\text{min}} \end{cases} \quad (10)$$

where g is the number of iterations; J_k^{max} and J_k^{min} are, respectively, the maximum and the minimal value of the l th quality constraint among the values of other available candidate solutions of the different clusters; ρ is the weight factor of the penalty; and, g_{max} is the maximum number of iterations. In addition, $g_{\text{current}}/g_{\text{max}}$ indicates that the penalty values are different at each iteration, and its aim is to be able to meet the constraint (10). If the objective function value is less than a specified maximum number of iterations, then the clustering operation ends.

The novel fuzzy C-means algorithm (FCML) that is based on Lagrange and K-means++ is shown in Algorithm 2.

Algorithm 2. The novel fuzzy C-means algorithm (FCML)Input: line segments (see Algorithm 1), K

Output: clustering results of line segments:

Procedure:

- (1) Randomly initialize the clusters membership values μ_{jk} in terms of line segments results;
- (2) Use Equation (8) to produce cluster centers;
- (3) Use Equation (6) to update membership values;
- (4) Use Equation (5) to calculate objective function values;
- (5) Use Equation (9) to repair error the rate of FCM, as well as to improve the global optimization and balance iterative hill-climbing;
- (6) Repeat steps 2–5 until J_m improves by less than the specified maximum number of iterations.

Finally, when line segments are achieved through the clustering operation, each cluster represents a different grouping result, which is made of the line segments in each cluster.

4.3. Trajectory Regression Clustering Based on the Least Squares Model

It is generally known that least squares regression (LSR) is a typical technique in statistics theory, which has been widely applied in the fields of pattern recognition, data mining, and machine learning [42–45], such as classification, clustering, and regression. It involves finding a hyperplane through a set of data points, while minimizing the objective function. In this paper, the LSR is employed to achieve a trajectory regression using line segments clustering results, as written in Equations (11) and (12).

$$\min_{\mathbf{w}, \mathbf{b}} \sum_{i=1}^n \left\| \sum_{k=1}^m \mathbf{A}_k \mathbf{x}_i^k - \mathbf{y}_i \right\|_2^2 \quad (11)$$

$$\frac{\partial \sum_{i=1}^n \left\| \sum_{k=1}^m \mathbf{A}_k \mathbf{x}_i^k - \mathbf{y}_i \right\|_2^2}{\partial \mathbf{A}^T} = 0 \Leftrightarrow \sum_{k=1}^n \left(\sum_{i=1}^m \mathbf{x}_i^k \right) \mathbf{A}_k = \sum_{i=1}^m \mathbf{x}_i^k \mathbf{y}_i \Leftrightarrow \mathbf{A}_k^T \mathbf{x}_i = \mathbf{x}_i \mathbf{y}_i \quad (12)$$

where $\mathbf{x}_i \in \mathbb{R}^{n \times m}$ is the clustering results, which denotes the k th cluster; and, m denotes the dimensionality of each cluster. $\mathbf{A}_k \in \mathbb{R}^{m \times k}$ denotes the regression matrix, $k (=1, 2, \dots, K)$ is the total number of clusters, and $\mathbf{y}_i \in \mathbb{R}^{n \times k}$ is a target matrix of \mathbf{x}_i . Equation (11) is used to minimize the J_m between the regressions $\mathbf{A}_k \mathbf{x}_i^k$ and \mathbf{y}_i , and is usually represented as a continuous vector. Equation (12) is employed to calculate the regression results. Therefore, if we need to obtain a regression curve, each trajectory clustering result can be put into Equations (11) and (12) in order to achieve trajectory regression clustering. The LSR-based regression clustering is shown in Algorithm 3.

Algorithm 3. LSR-based regression for clustering results from Algorithm 2

Input: clustering results (CR) of the line segments in Algorithm 2, number-order regression based on the least squares method

Output: regression trajectories

Procedure:

FOR 1 to K // K is number of clusters FOR 1 to n // n is number of taxi GPS data points OutputV(x, y) \leftarrow polyfit(CR $_K(n), K, number$)

// polyfit is the regression function based on LSR

// OutputV is the output function, and x and y denote the x axis and y axis, respectively

END FOR

END FOR

If the number of taxi GPS data points is n , the dimensionality is m , the number of clusters is K , and the number of iterations of FCML is g ; then, the time complexity of the presented method (FCML) is approximately equal to $O(\text{FCML}) = O(mn \log n) + O(gmnK \log n) + O(m^2nK) \approx O(n \log n)$ (if $n \gg m, n, K, g$), where $O(mn \log n)$ represents the complexity of the generated line segments, $O(gmnK \log n)$ is the complexity of FCML, and $O(m^2nK)$ is the complexity of the LSR computing. On the one hand, the time complexity of FCML is lower than the time complexity of the K-median with $O(n^2)$ or higher; on the other hand, the time complexity of FCML is approximately equal to the time complexity of K-means and FCM when the clustering algorithms are different (use K-mean and FCM to replace the improved FCM). Note that the complexity of FCML $O(\text{FCML})$ does not contain built-in functions of Matlab. The test results of time complexity are presented in Table 1. However, the run time of the K-means and the K-median are higher than that of FCML, because the K-means and K-median tend to get stuck in the local optimum.

Table 1. The average computational time (in minutes) of the clustering algorithms for real-world taxi GPS datasets.

The Number of Clusters	K-Means	K-Median	FCM	FCML
20	7.124884	6.835498	3.731286	3.728852
40	10.473148	10.035824	3.835722	3.824766
80	17.149494	16.434827	4.134353	4.018201
100	20.561356	19.632886	4.241650	4.117618

5. Experiment Results

In this paper, the experiment tests are presented in order to measure the performance of the cluster results in real-world taxi GPS datasets. The simulations are conducted in Matlab (v.2016b) on an Intel (R) Xeon (R) CPU E5-2658, computing at 2×2.10 GHz with 32 GB of RAM in Windows server 2008, which is running on a VMware-based cloud platform. Meanwhile, according to References [27,28], the *PBM-index* is superior to the common *DB-index* [46], *Dunn's index* [47], and *XB index* [24], in a measure of goodness of clustering on different partitions of a given dataset, and the *PBM-index* had been proposed as a measure of indication of the goodness/validity of a cluster solution in spatial data processing [48,49]. Therefore, the *PBM-index* is employed to compare the clustering performance between FCML and other partition-based clustering algorithms (K-means, K-median, and FCM (Fuzzy C-means)), as shown in Table 2 and Figure 11. A larger *PBM-index* value implies a better clustering result, which is defined as follows:

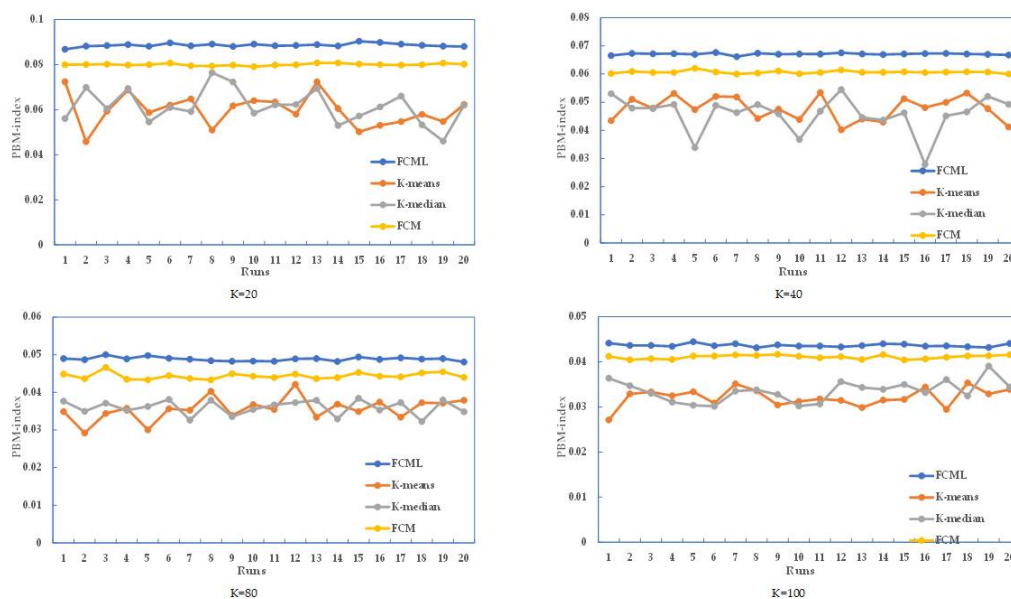
$$PBM(K) = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_K \right)^2 \quad (13)$$

where K is the number of clusters of line segments. Here, $E_K = \sum_{k=1}^K \sum_j^L \text{Hausd}(L_j, c_k)$ and $D_K = \max_{i,j=1}^K \{D(c_i, c_j)\}$. In addition, the *PBM-index* is regarded as an unsupervised clustering evaluation index and therefore a knowledge base about the true partitioning of the location data is not necessary. In other words, the map-based knowledge base of location information is not necessary, resulting in a trajectory that is not consistent with the map road. Moreover, the regression trajectory based on LSR is not consistent with urban roads. Furthermore, FCML is an unsupervised SLR-based regression clustering algorithm that does not employ a knowledge base as a support.

Table 2. The maximum (Max), mean, and minimum (Min) values of the PBM-index obtained by the K-means, K-median, and FCML for 20 different runs for four real-world taxi GPS datasets.

Values	K-Means	K-Median	FCM	FCML
K = 20				
Max	0.072370	0.076390	0.080722	0.090330
Mean	0.059763	0.061500	0.080010	0.088628
Min	0.045748	0.046031	0.079090	0.086792
K = 40				
Max	0.053466	0.054535	0.062047	0.066194
Mean	0.047736	0.045803	0.060685	0.067674
Min	0.040186	0.027937	0.060046	0.067124
K = 80				
Max	0.042144	0.038428	0.046632	0.050012
Mean	0.035606	0.035992	0.044379	0.048858
Min	0.029208	0.032240	0.043320	0.048086
K = 100				
Max	0.035346	0.039044	0.041666	0.044469
Mean	0.032129	0.033527	0.041100	0.043674
Min	0.027112	0.030160	0.040440	0.043137

The bold font indicates the best values for real-world taxi GPS data.

**Figure 11.** The PBM-index of the techniques (K-means, K-median, fuzzy C-means (FCM), and FCML) on different cluster numbers ($K = 20, 40, 80, 100$) for real-world taxi GPS datasets (Beijing, China).

The partitioning-based K-means, K-median, and FCM clustering algorithms in this paper are compared with FCML, and therefore the cluster numbers K of the clustering algorithms (K-means, K-median, FCM, and FCML) are set to 20, 40, 80, and 100, respectively; the number of iterations is set to 100 (terminating the local optimal of K-means and K-median), which is used as the termination condition of the cluster algorithms. In addition, the termination condition can be defined for $\frac{J(g+1)-J(g)}{J(g)} < \varepsilon$, where $J(g+1)$ is the next generation object value, $J(g)$ is the current generation object value, and ε is a given value (which is used to denote the minimum distance between line segments and the cluster centers of the line segments, e.g., $\varepsilon = 0.001$). The convergence of the clustering results is shown in Figure 12; the convergence values (J) in Figure 11 are normalized using the formula $\frac{J-J_{\min}}{J_{\max}-J_{\min}}$ in order to reach the same comparison standard, where J_{\max} and J_{\min} are the

maximum and minimum values of J (convergence values). In particular, when K-means, K-median, and FCM clustering algorithms are used, they are only employed in order to replace the improved FCM, while other operation conditions remain unchanged. In this paper, the two-order regression of the LSR is employed to produce trajectories in order to better explain tendency of the urban state changes without considering one-order (straight line) or other, as shown in Figure 13; in addition, other regression order numbers of LSR can also be set according to different requirements. The weight factor is set to $\rho = 0.3$ of the penalty of Lagrange, and $\alpha = 0.05$ and $\beta = 1$ values are employed in terms of the work in Reference [35]. In other words, when cluster results are obtained, the LSR can be employed in order to produce smoothness trajectories, which maybe be used to design and plan urban roads, support urban development, and identify traffic trends.

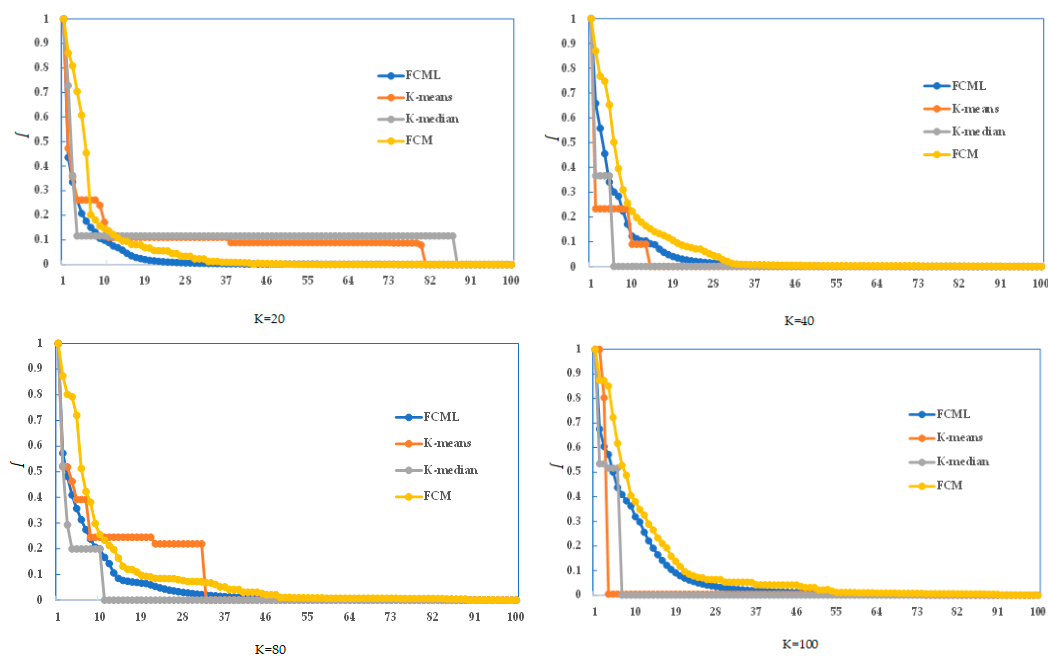


Figure 12. The figures plot the J obtained by the K-means, K-median, FCM, and FCML clustering algorithms, averaged over 20 independence trainings with different cluster numbers ($K = 20, 40, 80, 100$) on real-world taxi GPS datasets (Beijing, China).

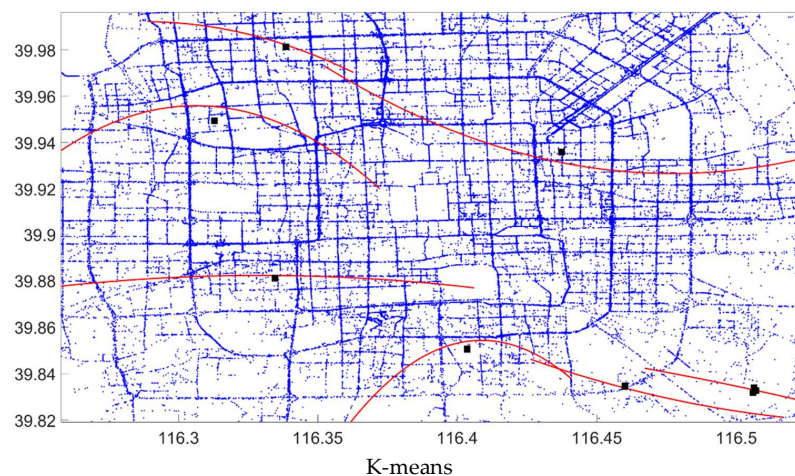


Figure 13. Cont.

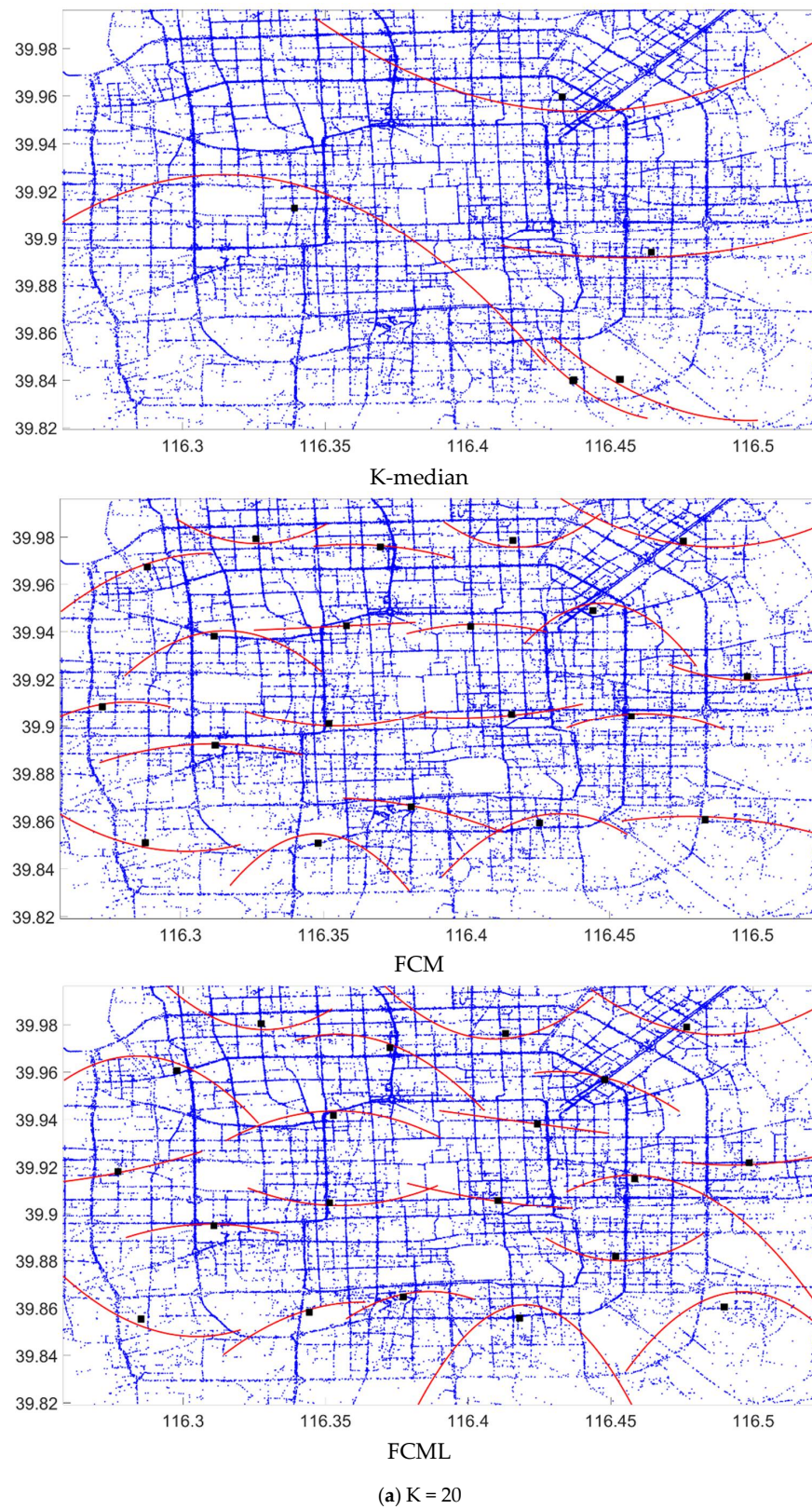


Figure 13. Cont.

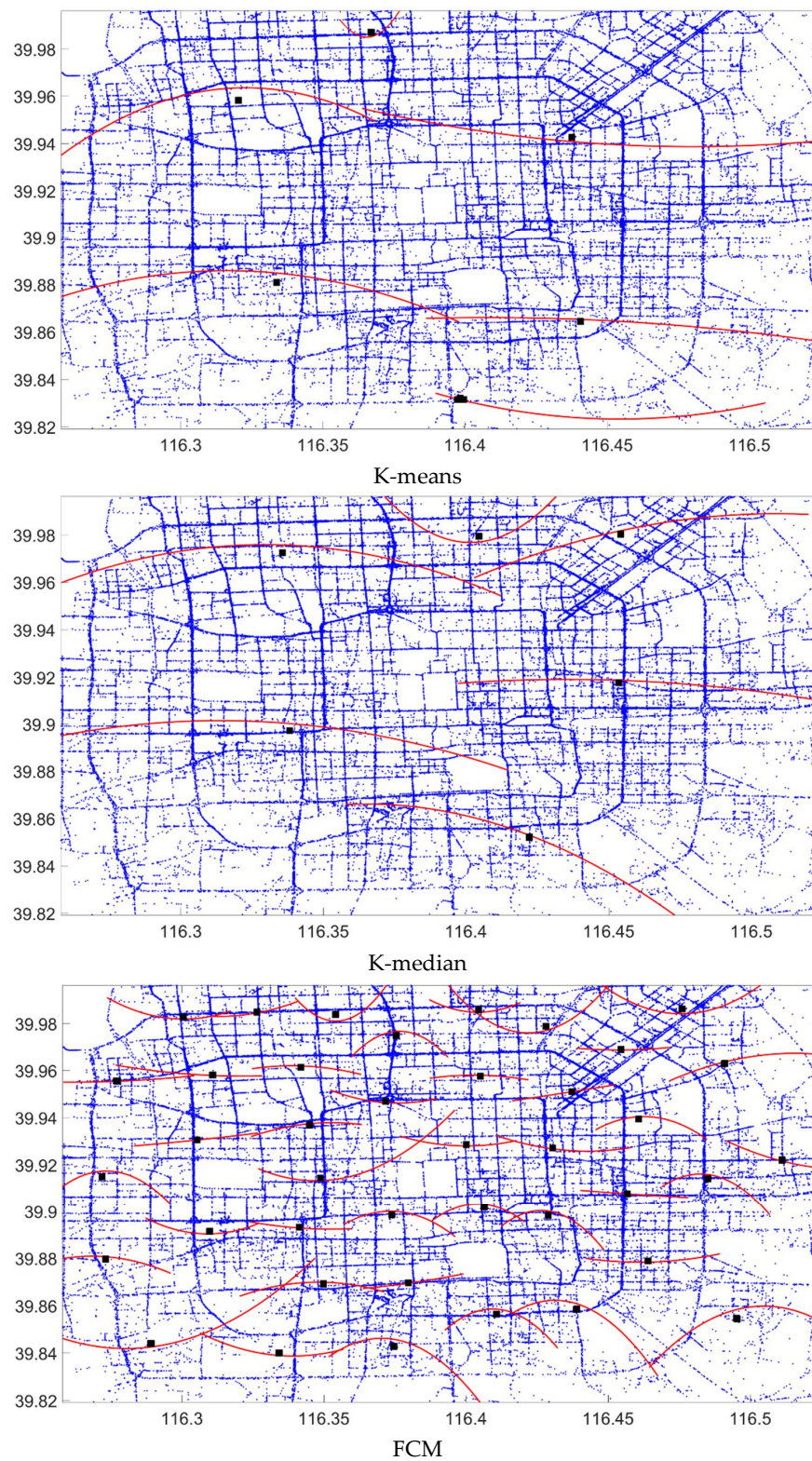
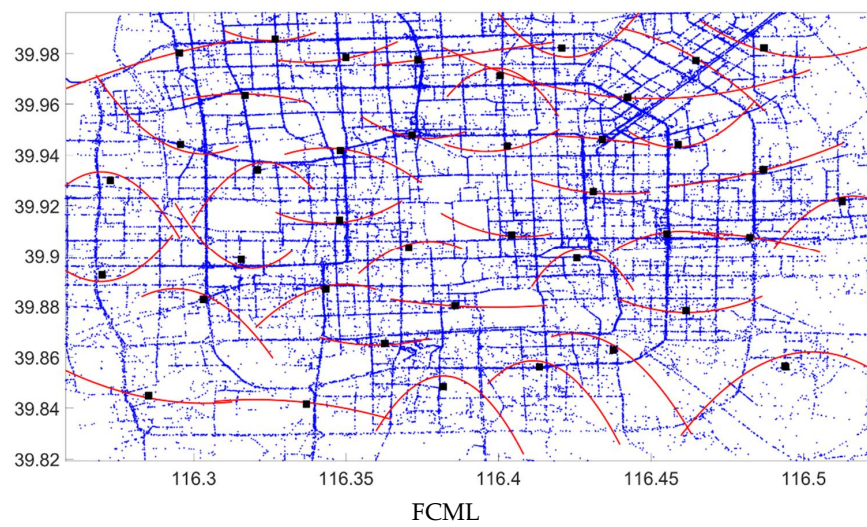


Figure 13. Cont.



(b) K = 40

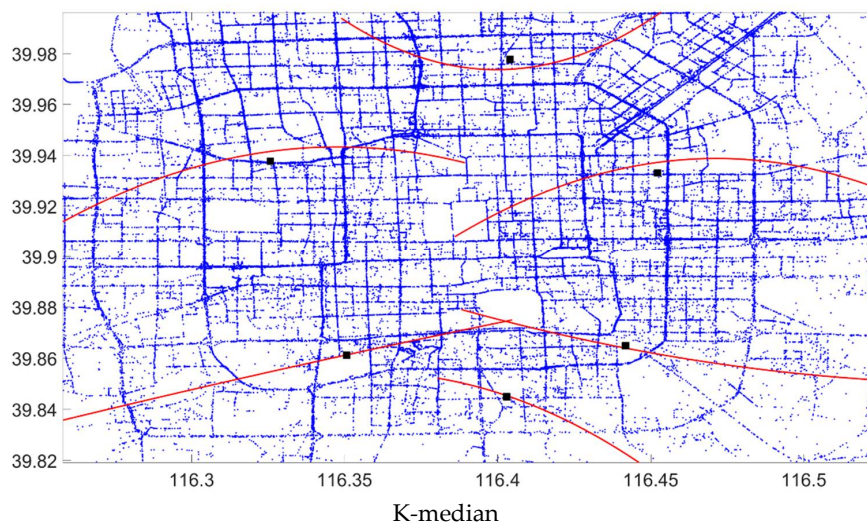
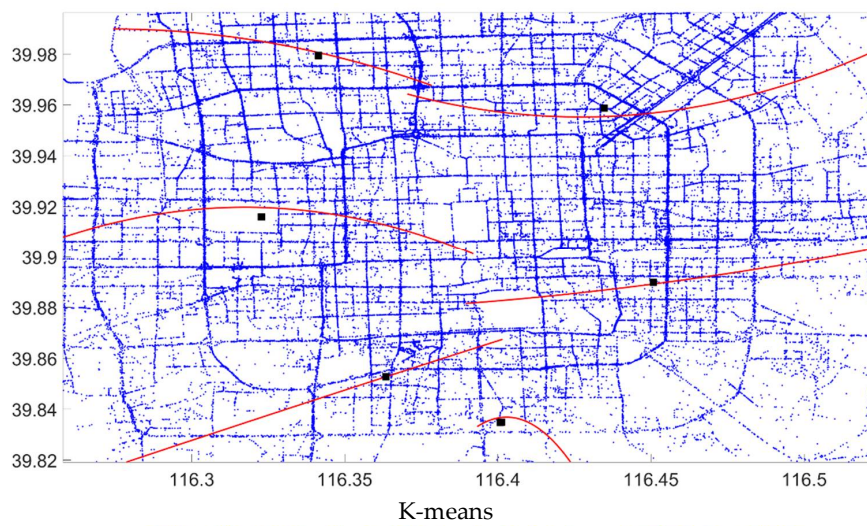


Figure 13. Cont.

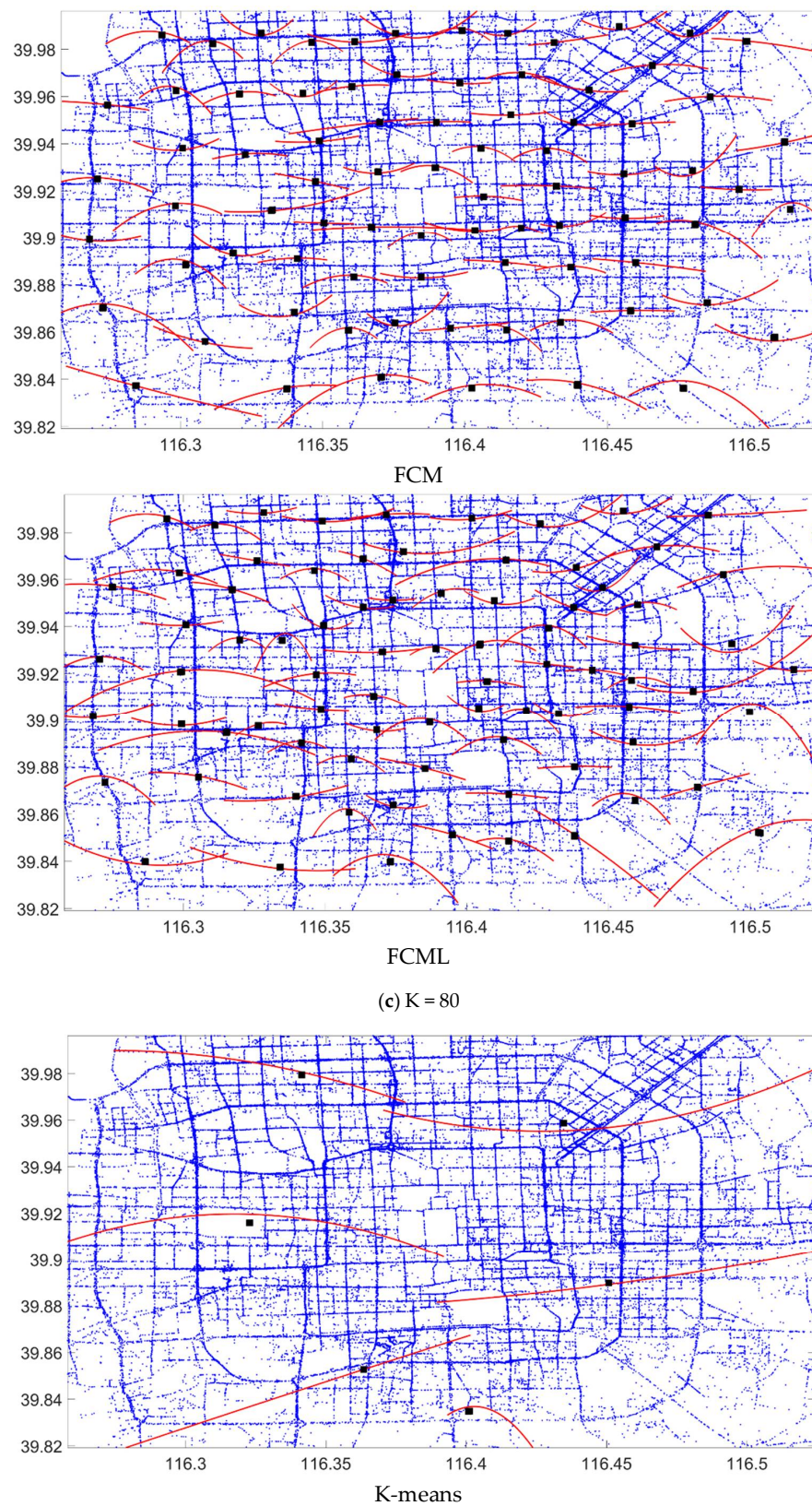


Figure 13. Cont.

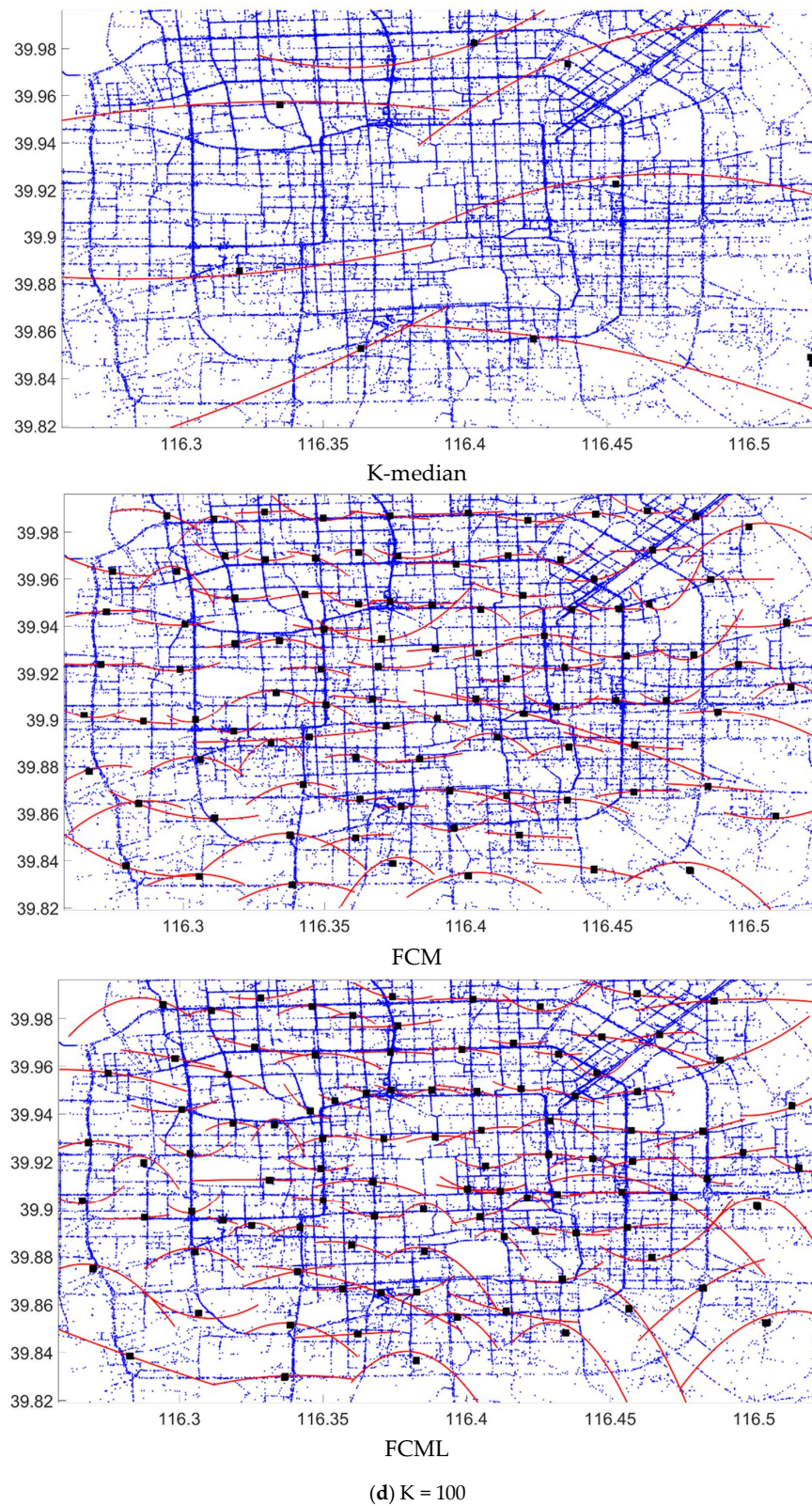


Figure 13. The trajectory regression clustering results of real-world taxi GPS data shown in Figure 2 using K-means, K-median, FCM, and FCML on the different cluster numbers: (a) K = 20, (b) K = 40, (c) K = 80, and (d) K = 100.

Table 2 and Figure 11 show the superiority of the FCML clustering algorithm, which produces a better value of the *PBM-index* than those of other partitioning-based clustering algorithms (K-means, K-median, and FCM). Moreover, FCM and FCML exhibited better stability than other algorithms on 20 different runs without suffering from randomness impacts, but the *PBM-index* values of FCML are obviously better than those of FCM, indicating that the Lagrange-based method that is used to improve FCM is effective.

In Figure 13, the red curves express regression trajectories of the clustering results for GPS data, which can be used to describe the tendency of the urban state changes; the black squares stand for cluster centers (line segments) of the clustering results, which can be used to explain hot points of roads (e.g., traffic flow and population aggregation segments) in general. Figure 13 demonstrates that FCML can obtain a better clustering and regression results, for example, when $K = 100$, cluster centers of the FCML is evenly distributed in the main road of the Beijing without deviating too much, and the smooth trajectories can be stretched around road and the cluster centers. In particular, if the cluster centers are linked together according to roads, the number of cluster centers are enough, and a map-based knowledge base is also supported, which can construct some real trajectories. However, the regression trajectory generated method in this paper is directly used to express the hidden information of GPS data and to explain state the changes of city without producing real trajectories.

Figure 12 shows that FCML obtains a better solution than other algorithms (K-means, K-median, and FCM), and the convergence process is very smooth, fast, and robust without getting stuck in the local optimum, and local information loss is also reduced. Meanwhile, the convergence of FCM is also smooth and robust (except before 10 iterations), but the convergence speed of FCML is faster than that of FCM; namely, FCML begins convergence at 20 iterations, and FCM begins convergence at about 30, 50 iterations when $K = 20$ and 40 , $K = 80$ and 100 , respectively. The regression trajectories can be used as a reference to establish city road planning and other fields of urban development. However, it should be noted that FCML is unsupervised clustering method for producing trajectories without a knowledge platform based on maps (e.g., Google maps). However, K-means and K-median clustering algorithms exhibit premature convergence and suffer from instability, resulting in the production of many empty clusters and the regression of long trajectories (see Table 3), as shown in Figure 13. In other words, a great quantity of line segments are gathered together, and therefore a cluster contains many line segments, resulting in a cost of computing time. For example, when $K = 20$, the number of line segments in each cluster is shown in Table 3; in addition, because K-means and K-median easily become stuck in the local optimum, a lot of time is used to gather line segments in order to find more line segments in a cluster, and when the number of clusters changes from 20 to 40, 80, 100, time consumption also increases as the number of clusters increases. However, the run time of FCML and FCM exhibit only a slight fluctuation in different cluster numbers. The test results of time consumption are shown in Table 1, revealing that the run time of FCML is slightly lower than that of FCM. Meanwhile, trajectories in Figure 13 can be used to explain the state of population migration around trajectories and is used as reference for road planning of the city.

Table 3. The number of line segments in each cluster using K-means, K-median, and FCML ($K = 20$).

Cluster	K-Means	K-Median	FCM	FCML
1	22	7367	1674	1185
2	0	0	1275	1262
3	0	0	587	1541
4	0	0	967	1629
5	0	0	1421	874
6	1145	0	1086	1344
7	5449	89	1325	680

Table 3. Cont.

Cluster	K-Means	K-Median	FCM	FCML
8	0	0	1140	1095
9	0	112	1295	1014
10	819	0	1925	1420
11	0	0	1487	908
12	0	0	993	1372
13	0	0	1190	946
14	3735	3695	1079	1189
15	0	12,522	684	1128
16	0	0	1533	1260
17	12,436	0	819	1267
18	0	0	1541	1475
19	179	0	976	1617
20	0	0	783	579

6. Conclusions

In this paper, we presented the FCML algorithm with the aim of achieving better performance of line segments clustering without getting stuck in the local optimum and losing more local information, in order to obtain a more effective regression trajectory. In the FCML algorithm, we first presented the new concept of trajectory and a new line segments generation method in order to reduce local information loss. A new Lagrange-based and Hausdorff-based distance K-means++ method was presented in order to improve the original fuzzy C-means clustering algorithm, which was used to avoid getting stuck in the local optimum, as well as to improve the convergence speed. In our improved fuzzy C-means method, the new Lagrange operator was used to adjust and control the similarity between line segments using Equation (5), as well as to achieve clustering operations. The Hausdorff-based K-means++ was employed to produce cluster centers. Finally, LSR was employed to achieve the regression of the clustering results and produce trajectories. In the experiments, we compared our method with three other clustering algorithms: K-means; K-median; and, FCM. The experimental results showed that FCML works better than K-means, K-median, and FCM.

However, our method requires the user to define the number of clusters in advance. Therefore, we will study an automatic generated method of the number of clusters that is based on PBM with a noise and density method in our future work. Meanwhile, when the FCML technique is used to support urban development, a large volume of GPS datasets must be analyzed; thus, we will study cloud-based analysis techniques in the future. In particular, when FCML is used in the context of real urban development, a knowledge platform that is based on maps needs to be established in future work.

Author Contributions: All authors contributed to this paper. Xianbing Zhou, Hongjiang Ma, and Fang Miao conceived the original idea for the study; Xianbing Zhou wrote the paper; Xiangbing Zhou, Hongjiang Ma, and Huaming Gong performed the experiments; and Xianbing Zhou and Huaming Gong analyzed the experiment results and revised the manuscript. All authors read and approved the submitted manuscript.

Acknowledgments: This paper was supported by the Research and Innovation Team of Universities and Colleges in Sichuan Province of China (15DT0039, 16DT0033), the Sichuan tourism youth expert training program (SCTYETP2017L02), and the Sichuan Science and Technology Program (18ZDYF3245, 2016GZ0140).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, Y.; Liu, Y.; Yuan, J.; Xie, X. Urban computing with taxicabs. In Proceedings of the 13th International Conference on Ubiquitous Computing, Beijing, China, 17–21 September 2011; pp. 89–98.
2. D’Andrea, E.; Marcelloni, F. Detection of traffic congestion and incidents from gps trace analysis. *Expert Syst. Appl.* **2017**, *73*, 43–56. [[CrossRef](#)]

3. An, S.; Yang, H.; Wang, J.; Cui, N.; Cui, J. Mining urban recurrent congestion evolution patterns from gps-equipped vehicle mobility data. *Inf. Sci.* **2016**, *373*, 515–526. [[CrossRef](#)]
4. Yang, Y.; Xu, Y.; Han, J.; Wang, E.; Chen, W.; Yue, L. Efficient traffic congestion estimation using multiple spatio-temporal properties. *Neurocomputing* **2017**, *267*, 344–353. [[CrossRef](#)]
5. Cui, J.; Liu, F.; Hu, J.; Janssens, D.; Wets, G.; Cools, M. Identifying mismatch between urban travel demand and transport network services using gps data: A case study in the fast growing chinese city of harbin. *Neurocomputing* **2016**, *181*, 4–18. [[CrossRef](#)]
6. Qu, M.; Zhu, H.; Liu, J.; Liu, G.; Xiong, H. A cost-effective recommender system for taxi drivers. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2014; pp. 45–54.
7. Cui, J.; Liu, F.; Janssens, D.; An, S.; Wets, G.; Cools, M. Detecting urban road network accessibility problems using taxi gps data. *J. Transp. Geogr.* **2016**, *51*, 147–157. [[CrossRef](#)]
8. Ferreira, N.; Poco, J.; Vo, H.T.; Freire, J.; Silva, C.T. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2149–2158. [[CrossRef](#)] [[PubMed](#)]
9. Kharrat, A.; Popa, I.S.; Zeitouni, K.; Faiz, S. Clustering algorithm for network constraint trajectories. In *Headway in Spatial Data Handling*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 631–647.
10. Lee, J.-G.; Han, J.; Whang, K.-Y. Trajectory clustering: A partition-and-group framework. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, Beijing, China, 12–14 June 2007; pp. 593–604.
11. Deng, Z.; Hu, Y.; Zhu, M.; Huang, X.; Du, B. A scalable and fast optics for clustering trajectory big data. *Clust. Comput.* **2015**, *18*, 549–562. [[CrossRef](#)]
12. Han, B.; Liu, L.; Omiecinski, E. Road-network aware trajectory clustering: Integrating locality, flow, and density. *IEEE Trans. Mob. Comput.* **2015**, *14*, 416–429.
13. Lou, Y.; Zhang, C.; Zheng, Y.; Xie, X.; Wang, W.; Huang, Y. Map-matching for low-sampling-rate gps trajectories. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009; pp. 352–361.
14. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, X.; Sun, G.-Z. An interactive-voting based map matching algorithm. In Proceedings of the 2010 Eleventh International Conference on Mobile Data Management (MDM), Kansas City, MO, USA, 23–26 May 2010; pp. 43–52.
15. Ciscal-Terry, W.; Dell’Amico, M.; Hadjidimitriou, N.S.; Iori, M. An analysis of drivers route choice behaviour using gps data and optimal alternatives. *J. Transp. Geogr.* **2016**, *51*, 119–129. [[CrossRef](#)]
16. Luo, T.; Zheng, X.; Xu, G.; Fu, K.; Ren, W. An improved dbscan algorithm to detect stops in individual trajectories. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 63. [[CrossRef](#)]
17. Mai, G.; Janowicz, K.; Hu, Y.; Gao, S. Adcn: An anisotropic density-based clustering algorithm for discovering spatial point patterns with noise. *Trans. GIS* **2018**, *22*, 348–349. [[CrossRef](#)]
18. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
19. Jain, A.K. Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
20. Lv, M.; Chen, L.; Xu, Z.; Li, Y.; Chen, G. The discovery of personally semantic places based on trajectory data mining. *Neurocomputing* **2016**, *173*, 1142–1153. [[CrossRef](#)]
21. Lecue, F.; Mehadjiev, N. Seeking quality of web service composition in a semantic dimension. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 942–959. [[CrossRef](#)]
22. Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
23. Bahmani, B.; Moseley, B.; Vattani, A.; Kumar, R.; Vassilvitskii, S. Scalable k-means++. *Proc. VLDB Endow.* **2012**, *5*, 622–633. [[CrossRef](#)]
24. Pal, N.R.; Bezdek, J.C. On cluster validity for the fuzzy c-means model. *IEEE Trans. Fuzzy Syst.* **1995**, *3*, 370–379. [[CrossRef](#)]
25. Henrikson, J. Completeness and total boundedness of the hausdorff metric. *MIT Undergrad. J. Math.* **1999**, *1*, 69–80.
26. Bandyopadhyay, S.; Maulik, U. An evolutionary technique based on k-means algorithm for optimal clustering in rn. *Inf. Sci.* **2002**, *146*, 221–237. [[CrossRef](#)]

27. Pakhira, M.K.; Bandyopadhyay, S.; Maulik, U. Validity index for crisp and fuzzy clusters. *Pattern Recognit.* **2004**, *37*, 487–501. [CrossRef]
28. Maulik, U.; Bandyopadhyay, S. Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1650–1654. [CrossRef]
29. Real-World Taxi-Gps Data Sets. Available online: <https://github.com/bigdata002/Location-data-sets> (accessed on 10 October 2017).
30. Zhou, X.; Gu, J.; Shen, S.; Ma, H.; Miao, F.; Zhang, H.; Gong, H. An automatic k-means clustering algorithm of gps data combining a novel niche genetic algorithm with noise and density. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 392. [CrossRef]
31. Lu, M.; Liang, J.; Wang, Z.; Yuan, X. Exploring od patterns of interested region based on taxi trajectories. *J. Vis.* **2016**, *19*, 811–821. [CrossRef]
32. Spaccapietra, S.; Parent, C.; Damiani, M.L.; de Macedo, J.A.; Porto, F.; Vangenot, C. A conceptual view on trajectories. *Data Knowl. Eng.* **2008**, *65*, 126–146. [CrossRef]
33. Luo, C.; Junlin, L.; Li, G.; Wei, W.; Li, Y.; Li, J. Efficient reverse spatial and textual k nearest neighbor queries on road networks. *Knowl.-Based Syst.* **2016**, *93*, 121–134. [CrossRef]
34. Chang, C.; Zhou, B. Multi-granularity visualization of trajectory clusters using sub-trajectory clustering. In Proceedings of the IEEE International Conference on Data Mining Workshops, Miami, FL, USA, 6–9 December 2009; pp. 577–582.
35. Li, Y.; Bandar, Z.A.; McLean, D. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.* **2003**, *15*, 871–882.
36. Scim, S.; Lsmailm, A. Means-type algorithm: A generalized convergence theorem and characterization of local optimality. *IEEE. Trans. Pattern Anal.* **1984**, *6*, 81–87.
37. Cox, E. *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*; Elsevier: Amsterdam, The Netherlands, 2005; pp. 421–481.
38. Saha, A.; Das, S. Axiomatic generalization of the membership degree weighting function for fuzzy c means clustering: Theoretical development and convergence analysis. *Inf. Sci.* **2017**, *408*, 129–145. [CrossRef]
39. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenum Press: New York, NY, USA, 1981; pp. 203–239.
40. Ding, Y.; Fu, X. Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing* **2016**, *188*, 233–238. [CrossRef]
41. Mukhopadhyay, A.; Maulik, U. Towards improving fuzzy clustering using support vector machine: Application to gene expression data. *Pattern Recognit.* **2009**, *42*, 2744–2763. [CrossRef]
42. Yuan, H.; Zheng, J.; Lai, L.L.; Tang, Y.Y. A constrained least squares regression model. *Inf. Sci.* **2018**, *429*, 247–259. [CrossRef]
43. Liu, Y.; Zhang, S. Fast quantum algorithms for least squares regression and statistic leverage scores. *Theor. Comput. Sci.* **2017**, *657*, 38–47. [CrossRef]
44. Chen, K.; Lv, Q.; Lu, Y.; Dou, Y. Robust regularized extreme learning machine for regression using iteratively reweighted least squares. *Neurocomputing* **2017**, *230*, 345–358. [CrossRef]
45. Gui, J.; Sun, Z.; Ji, S.; Tao, D.; Tan, T. Feature selection based on structured sparsity: A comprehensive study. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1490–1507. [CrossRef] [PubMed]
46. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE. Trans. Pattern Anal.* **1979**, 224–227. [CrossRef]
47. Dunn, J.C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. Cybern.* **1973**, *3*, 32–57. [CrossRef]
48. Chang, D.-X.; Zhang, X.-D.; Zheng, C.-W. A genetic algorithm with gene rearrangement for k-means clustering. *Pattern Recognit.* **2009**, *42*, 1210–1222. [CrossRef]
49. Bandyopadhyay, S.; Maulik, U.; Mukhopadhyay, A. Multiobjective genetic clustering for pixel classification in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 1506–1511. [CrossRef]

