International Journal of Geo-Information

MDPI

Article

Clustering Methods Based on Stay Points and Grid Density for Hotspot Detection

Xiaohan Wang [1,2], Zepei Zhang [2] and Yonglong Luo [1,2,*]

[1] School of Computer and Information, Anhui Normal University, Wuhu 241002, China; xiaohanwang@ahnu.edu.cn
[2] Anhui Provincial Key Laboratory of Network and Information Security, Wuhu 241002, China; zhangzp12@lenovo.com
* Correspondence: ylluo@ustc.edu.cn

Abstract: With the widespread use of GPS equipment, a large amount of mobile location data is recorded, and urban hotspot areas extracted from GPS data can be applied to location-based services, such as tourist recommendations and point of interest positioning. It can also provide decision support for the analysis of population migration distribution and land use and planning. However, taxi GPS location data has a large amount of data and sparse points. How to avoid the influence of noise and efficiently detect hotspots in cities have become urgent problems to be solved. This paper proposes a clustering algorithm based on stay points and grid density. Firstly, a filtering pre-processing algorithm using stay points classification and stay points thresholds is proposed, so the influence of stop points is avoided. Then, the data space is divided into rectangular grid cells; each grid cell is determined to be a dense or non-dense grid according to the defined density threshold, and the cluster boundary points and noise points are judged in the non-dense grid cells to avoid normal sampling points being treated as noise. Finally, the associated dense grids are connected into clusters. The sampling points mapped to the grid cells are the elements in the clusters. Our method is more efficient than the DBSCAN algorithm because the grid cells are calculated. The superiority of the proposed algorithm in terms of clustering accuracy and time efficiency is verified in the real data set compared to traditional algorithms.

Keywords: clustering; stay points; grid density; hotspot detection

Citation: Wang, X.; Zhang, Z.; Luo, Y. Clustering Methods Based on Stay Points and Grid Density for Hotspot Detection. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 190. https://doi.org/10.3390/ijgi11030190

Academic Editor: Wolfgang Kainz

Received: 6 December 2021
Accepted: 7 March 2022
Published: 11 March 2022

1. Introduction

With the development of refined urban management and the government's emphasis on improving the living and working environment, the in-depth application of smart cities will enter a new stage of development. Urban hotspot area mining is an important issue in the construction of smart cities. The dynamic changes in urban hotspot areas [1], combined with its land use semantic information, can be used to reveal the functionality of urban land use [2,3]. Various types of GPS sensors collect information such as latitude and longitude coordinates and time of mobile users to form GPS location datasets, such as social media check-in locations [4], GPS traffic track locations [5], smart card recording locations [6–8] and mobile phone locations [9], etc. For massive GPS geographic location data, a simple location probability model [10] and visual analysis technology [11] used to mine clusters of identifying hotspots are easily affected by noise data. How to efficiently deal with massive location data information is still a problem that needs to be solved urgently.

Existing research on urban hotspot mining mainly includes high-resolution satellite image recognition, complex networks, and statistical analysis. Traditional satellite recognition methods using high-resolution satellite images to measure relevant information in urban areas require more time consumption [12]. These methods are expensive, professional, and difficult to promote and apply, so they cannot meet the requirements of timeliness and

low cost. Another method is the complex network method, which mainly uses the theory of complex networks to describe and analyze hotspot information [13]. Some statistical analysis and pattern mining methods have also been used to identify hotspots [14–16].

Urban hotspot mining can also be implemented using clustering. Clustering-based methods for hotspot area mining do not require manual intervention, which can simplify complexity. For example, Junghoon et al. used the K-means cluster analysis on the location data of taxis in Jeju area to obtain hotspots, and recommended the hotspots to taxi drivers [17]. Thuillier et al., based on a large number of phone records data, obtained urban hotspots through clustering to determine the way people travel [18]. Clustering is a machine learning technique that groups data [19], and it is widely used in pattern recognition [20], decision support [21], image processing [22–24], data mining [25], genetic testing [26], etc. The basic principle of the clustering is to divide the original data into multiple disjoint areas according to the principle that the distance between elements in the cluster is relatively small and the distance between the clusters is relatively large.

However, the clustering algorithm involves a large number of sampling point distance judgment calculations when processing massive GPS location data, which seriously affects time performance and processing efficiency. In addition, the existing clustering analysis for urban hotspot areas does not combine actual factors. For example, sparsely distributed sampling points are generated due to the remote sampling records of taxis. Existing clustering algorithms are sensitive and the accuracy of clustering results is greatly affected by sparsely distributed sampling points, which may lead to unreliable estimation of urban hotspot areas. An important issue is how to reduce the impact of noisy data and improve the accuracy of clustering while completing efficient cluster mining for spots.

To solve the problem of existing clustering methods not efficiently completing the clustering task and their sensitivity to sparse noise points in the data set, this paper proposes clustering methods based on stay points and grid density, including stay point filtering, grid mapping, boundary point judgments and dense grid clustering. The specific contributions are as follows:

(1) In view of the large number of stay points in the taxi position data set, this paper proposes a filtering pre-processing based on stay point classification and stay point thresholds, which can avoid the grid density in some areas being too high due to vehicle stay events.
(2) The original position data space is divided into rectangular grid cells in the process of grid mapping and boundary point determination, and whether each grid cell is a dense grid is determined according to the defined density threshold; we determine cluster boundary points and noise points in non-dense grid cells to avoid identifying normal data as noise in order to process noise data more accurately.
(3) In view of the low efficiency of the existing clustering methods when processing large amounts of data, this paper connects the associated dense grid cells to form clusters. Since clustering is oriented to grid cells, it is more efficient than traditional algorithms.
(4) Finally, the experiments in the real data sets verify that the algorithm reduces the time cost of clustering.

2. Related Works

At present, there are three types of clustering methods in domestic and foreign research.: partition-based methods, hierarchical clustering, and density clustering. Density-based clustering does not need to define the number of clusters in advance, and it can identify clusters of different shapes, which has a good effect in finding high-density regions. The basic principle of the density-based method is that when the density of the neighborhood of a data point exceeds a certain threshold, it continues to search the neighborhood for the sampling points in the neighborhood, and finally the data points in a nearby range is a cluster. This type of method defines two parameters, the maximum radius of the adjacent area and the density of the adjacent area.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a classic density-based algorithm. Since the DBSCAN based method can accurately extract high-density points in the location data set, it is effectively used in hotspot mining. In order to explore the impact of clustering on the road network structure [27], Schoier used the classic DBSCAN clustering algorithm to perform a cluster analysis on the urban area of Trieste (Italy) to understand the structure of the road network from the "dense" areas of the location point. However, whether the clustering results of this algorithm are meaningful to real users has not been evaluated systematically. In response to this problem, Zhou proposed an improved density and connection-based clustering algorithm for mining hotspots that are meaningful to individuals [28], and the author proves that results from the algorithm has practical meaning by collecting real user data, but this method only considers the spatial size and ignores the time series features. Hwang took the spatial-temporal characteristics into consideration and used linear interpolation to fill in the position points that did not meet the spatial density and duration measurement criteria. A DBSCAN spatial clustering algorithm considering time criteria and intervals was proposed to detect urban hotspots [29]. When calculating the density of GPS points, many clusters mainly consider the number of GPS points within a given distance rather than their corresponding characteristics. Luo et al. used a Gaussian function to measure the density by the number of points within a certain distance from the current point instead of the density calculation method of the current point in the DBSCAN algorithm. A DBSCAN clustering algorithm based on mixed features was proposed [30] which first defined the new concept of mobility; clustering area should have lower mobility and higher GPS point density, each location point is affected by the interaction with other points, and more accurate clustering results can be obtained.

As an emerging data mining technology, clustering methods can use machine processing to avoid cumbersome and inaccurate manual statistical data. However, the above-mentioned traditional density-based clustering method directly performs clustering tasks on data points. It requires a large number of calculations and has low algorithm execution efficiency when processing a large amount of location sampling point data.

We can use grid clustering to solve this problem. Because the whole data space is divided into grid cells according to the side length, the cell processed by the algorithm is the divided grid cell instead of individual data, so the clustering efficiency can be improved [31]. At present, there have been studies using grid optimization density clustering algorithms, which can improve the efficiency of the algorithm and screen for sparse noise data. In terms of processing noisy data, Zhao et al. proposed a grid growth and improved density clustering method [31]. The sparse grid area is removed as outlier noise data in the algorithm, which enhances the algorithm's noise processing ability and is suitable for large geographic spatial data, with a competitive advantage in running time [31].

However, the above method does not further filter the sparse grid when processing noise data, and the sparse grid may serve as the boundary grid of the cluster. Direct removal affects clustering results and the noise data judgment is inaccurate, which may result in the problem of reduced data availability.

Finally, the above improved DBSCAN algorithm [27–29] does not consider the stay sampling points in the taxi data set in the processing, which will affect the accuracy of density clustering methods. In order to solve the above problems, this paper proposes a clustering method based on stay points and grid density. Firstly, the algorithm of this paper adopts the pre-processing step of stay point filtering to reduce the impact of stay points on clustering. Secondly, the clustering method based on stay points and grid density judges the boundary points in the sparse grid cell according to the preset density threshold; this method further refines the clustering boundary points by performing grid translation, and finally the sparse grid cell that is not part of the boundary is judged as noise data, which achieves more accurate judgment of the data. This method clusters by grid, which significantly enhances the execution efficiency of the algorithm.

3. Related Definitions

First, the definitions of some concepts in this chapter are given as follows.

Definition 1. ***(grid cell)*** *Suppose an n-dimensional space D is given, and we divide each dimension $D_1, D_2, \ldots, D_n$ in space D into $m_1, m_2, m_3, \ldots m_n$; each of grid cell has the same side length. The space D is divided into $m_1 * m_2 * m_3 * \ldots * m_n$ grid cells. Each grid cell d_i in space D can be expressed as follows.*

$$d_i = \{d_{i_1}, d_{i_2}, d_{i_3}, \ldots, d_{i_n}\} \tag{1}$$

where $d_{i_j} = [l_{i_j}, h_{i_j})$ is the interval of the grid cell d_i in the D_j dimension and satisfying $1 \leq j \leq n$, l_{i_j}, h_{i_j} are the left and right endpoints of the interval, and the length of the interval is the side length of the grid cell. Since this paper is about the cluster analysis of the sampling points by taxi positions data, the data source here is on a two-dimensional plane, so the coordinate space dimension n is 2 and the grid cells can be visually expressed as a square grid.

Definition 2. ***(density of grid cells)*** *After dividing the space, the number of data points falling into a grid cell is the density of the grid cell. Let the input data point set be as follows.*

$$V = \{v_1, v_2, v_3, \ldots v_n\} \tag{2}$$

where $v_i = \{v_{i1}, v_{i2}, v_{i3}, \ldots v_{in}\}$, and v_{ij} is the component of the data point v_i in the data point set V in the D_j dimension. If a data point v_i falls in a grid cell d_i on the D_j dimension, the condition needs to be met as follows.

$$l_{i_j} \leq v_{ij} < h_{i_j}, \ (1 \ \leq \ j \ \leq \ n) \tag{3}$$

l_{i_j},h_{i_j} are the minimum and maximum values of the interval, respectively. If the data point v_i falls in a grid cell d_i in n dimensions, the grid cell density count is increased by one. In the two-dimensional location sampling point set, V is usually a set of latitude and longitude points, so it is only necessary to determine whether each sampling point falls within the interval of the corresponding grid cell on the two indexes of longitude and latitude to perform density counting.

Definition 3. ***(grid center point)*** *The grid center point refers to the center point of each grid cell, and the center point of a grid cell d_i is as follows:*

$$gridc_i = (gridc_{i1}, gridc_{i2}, gridc_{i3}, \ldots, gridc_{ij}, \ldots gridc_{in}) \tag{4}$$

where $gridc_{ij}$ is the mathematical center point of grid cell d_i in D_j dimension, and its calculation formula is as follows:

$$gridc_{ij} = \frac{l_{i_j} + h_{i_j}}{2}, \ (1 \ \leq \ j \ \leq \ n) \tag{5}$$

where l_{i_j} , h_{i_j} are the minimum and maximum values of the interval, respectively. Among them, the dimension of grid data is two in this paper, namely longitude and latitude. Therefore, the center point of grid cell d_i is $gridc_i = (longitude, latitude)$, where the calculation method of longitude is the average of the minimum and maximum longitude of the grid cell. The same is true for latitude calculations.

Definition 4. ***(data center point)*** *It refers to the center position of the data points contained in the grid cell. There are k data points in a grid cell d_i expressed as $V = \{v_1, v_2, v_3, \ldots v_k\}$, and then the grid cell data center point is calculated by Formula (6).*

$$datac_i = (datac_{i1}, datac_{i2}, datac_{i3}, \ldots, datac_{ij}, \ldots, datac_{in}) \tag{6}$$

Among them, $datac_{ij}$ is the arithmetic mean value of the projection components of the k data points V on the Dj dimension, and is calculate by Formula (7).

$$datac_{ij} = \frac{v_{1j} + v_{2j} + v_{3j} + \ldots + v_{kj}}{k}, \ (1 \leq j \leq n) \tag{7}$$

According to the grid center point above, the dimension n used in this paper is two, so the formal representation of the data center point is still $datac_i = (longitude, latitude)$, where the longitude calculation method is the average of longitude with all data points in the grid cell, and latitude is calculated in the same way.

Definition 5. ***(directly associated grid cells)*** *Two grid cells have intersections in at least one dimension, and these two grid cells are said to be directly related.*

4. Clustering Method Based on Stay Point and Grid Density

The clustering method based on stay point and grid density proposed in this paper mainly includes four parts, namely stay point filtering, grid mapping, boundary point judgment and dense grid clustering. Firstly, we identify and filter the stay points of the original location data. In this stage, the predefined taxi events and the stay time are used to classify the stay points, and the different stay point thresholds are used to filter the data set. Through experiments, we can get the parameter settings of the classification and threshold value of the stay point applicable to this data set. Secondly, grid mapping is performed on the filtered position data. At this stage, the preset grid cell side length and density threshold are used to divide the original data space and the position sampling points are mapped into the corresponding grid to determine the dense grid cell set. Furthermore, we determine the cluster boundary points and noise points in non-dense grid cells. Finally, we use all the directly related dense grid cells form multiple clusters to complete the grid clustering.

4.1. Stay Point Filtering

In a real situation, there will be extra stops such as staying at some places to wait for guests, stopping at intersections, etc. However, the position sensor still regularly uploads GPS information, resulting in too many sampling points in the area, and these stay points lead to inaccurate clustering results. Many clustering algorithms for detecting hotspot regions in the existing literature do not consider the judgment of stay points [27–29]. In response to this problem, this paper proposes a pre-treatment method for stay point filtering.

According to the common taxi events and their stay time, this paper proposes five kinds of stay events as shown in Table 1. The stay time is represented by Δt. According to the defined stay events, the stay points in the original sampling points can be classified and extracted.

Table 1. Stay events and stay time of points.

Stay Events	Stay Time
Waiting for traffic lights	$\Delta t \leq 1$ min
Taxi pick-up and drop-off	1 min $< \Delta t \leq 3$ min
Traffic jam	3 min $< \Delta t \leq 30$ min
Business suspension	30 min $< \Delta t \leq 120$ min
Business breaks	120 min $< \Delta t \leq 600$ min

Due to accuracy errors in GPS positioning, even the latitude and longitude uploaded twice at the same location may be different. Therefore, this paper proposes the threshold of stay points to solve this positioning error.

When the threshold value of the stay point from the latitude angle is 0.0001, 0.001, 0.01, according to the latitude and longitude network maps around the world, the interval length of latitude 1° is equal (because the length of all meridians is equal) and the relevant

distance standard is calculated according to the latitude and longitude. The corresponding actual distances represent 0.0111 km, 0.111 km, and 1.11 km, respectively.

The data pre-processing method for filtering the different types of stay points of the taxi GPS data set according to the stay events is closer to the real world; the accuracy of the method is higher, the flexibility is stronger and it can be personalized. For example, we can extract and filter sampling points with a sampling interval of less than one minute and the latitude and longitude basically unchanged from the data set to solve the problem of inaccurate clustering results due to the high density of grid cells caused by waiting for traffic lights.

4.2. Grid Mapping

After filtering out the pre-processing stage of stay points, it is necessary to perform the grid mapping. The grid size affects the clustering results; as grid size increases, the accuracy decreases. The specific setting needs to be combined with the actual application scenario. The paper mainly sets the grid length of the corresponding size in combination with different data sets.

The main task of grid mapping is to mesh the original position points and calculate the density of the corresponding grid cell. Firstly, we find the minimum point of latitude and longitude in the data set as the origin and divide the grid cell into the entire data space according to the predefined grid cell side length. Secondly, all the grids are screened to determine whether they are dense grids. According to the GPS coordinates of the original sampling points, it is determined which specific grid they belong to, and the number of sampling points is calculated to determine the grid cell density. A cell is a dense grid cell if the number of data points in the grid is greater than the density threshold; otherwise, it is a non-dense cell.

4.3. Boundary Point Judgment

Many existing clustering methods directly set the non-dense cells as noise points, which causes many boundary points to be regarded as noise, and leads to inaccuracy of the clustering results. In this paper, we further refine the non-dense grid cells to find cluster boundary data and sparse noise data.

In this paper, the boundary point judgment process divides non-dense cells into two types. One type is a non-dense grid, with any dense grid cell directly set as a dense grid. The other type are grids that are not directly associated with dense cells. This paper proposes a grid cell center translation method. This method translates a grid that does not satisfy the density threshold and is not associated with dense grid cells; then, we can distinguish cluster boundary points and noise points by the density of the new grid.

When the grid cell center is shifted, we move the grid center point to the data center point and keep the grid side length unchanged; then, we recalculate the new grid cell density. If the value is greater than the density threshold, the new grid cell is set for dense grids. If the density threshold is still not met, the grid cell is added to the noise grid cell set. The calculation of grid center and data center points is shown in Figure 1, and the translation process of grid cell centers is shown in Figure 2.

In Figure 1, the solid line box is a grid cell and the density of the grid cell is three; there are three data sampling points. The calculation method of the grid center point is $[(0 + 1)/2, (0 + 1)/2] = (0.5, 0.5)$. The data center point is the average of the horizontal and vertical coordinates of the three black data sampling points.

Figure 2 shows the moving process of the grid cell in this paper. Firstly, the grid cell centered on the grid center p is the initial grid. At this time, there are three data sampling points in the grid, and the density of grid cells is three. In the process of judging the boundary points, it is necessary to move the grid center to point p', which is the data center point of the grid. The new grid constructed at this time includes four new red sampling points, so the grid cell density is seven.

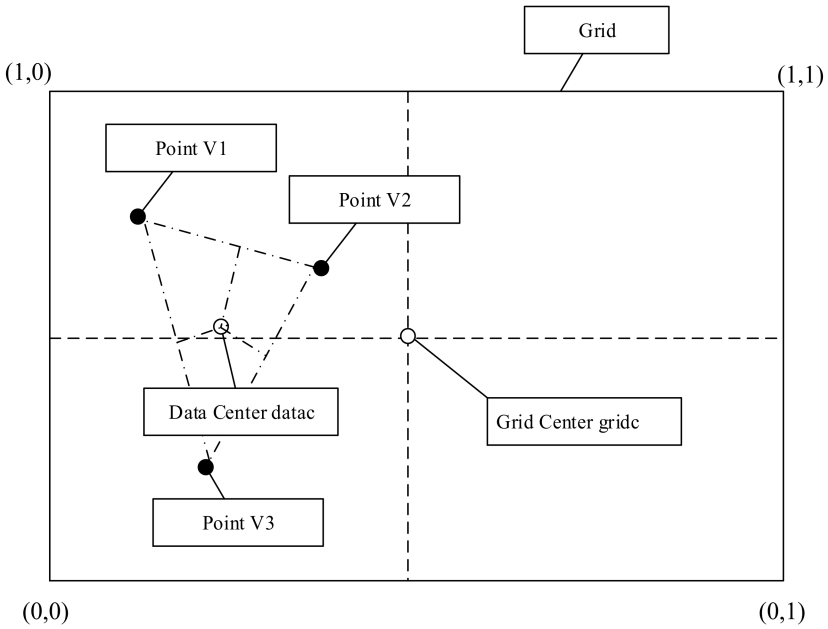


Figure 1. Data center point and grid center point.

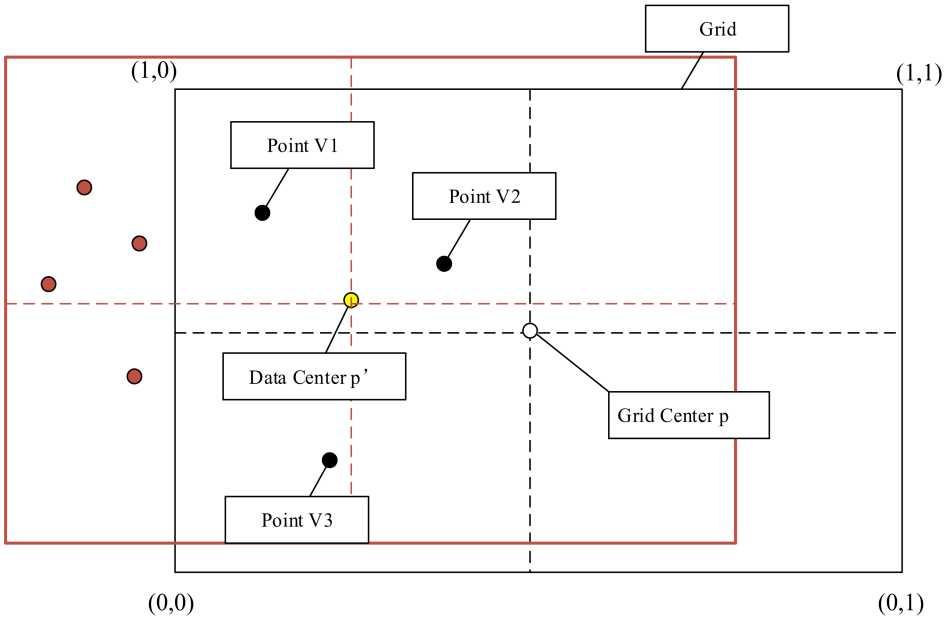


Figure 2. Grid center (p → p′) translation process.

Moving the grid center from the grid center point to the data center point helps to correctly identify the dense grid and avoid identifying dense points as sparse grids and deleting them. As shown in the Figure 2, the original grid density is three, and the density after moving is seven. If the density threshold is five, the process can avoid the screening of data points in the figure.

After the above grid movement process, we recalculate the density of the new grid cell. If the value is greater than the density threshold, the new grid cell will be set to dense, otherwise it will be set to noise grid. When the cluster boundary points and noise points are judged, so we can get all points in the noise grid are noise data. Algorithm 1 describes the judgment steps of cluster boundary points in detail. An example of the algorithm execution process is shown in Figure 3.

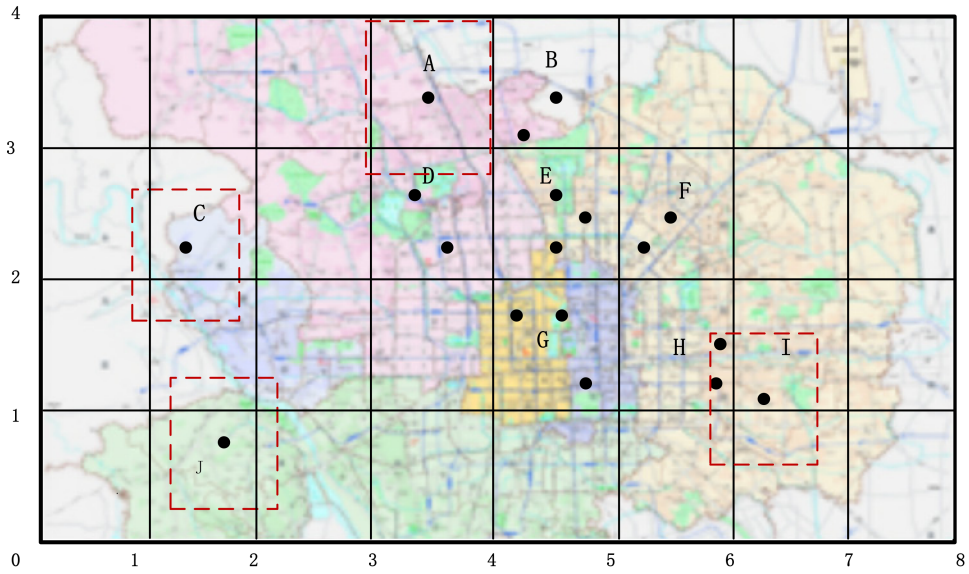


Figure 3. Judgment of cluster boundary points.

Algorithm 1. Cluster boundary judgment algorithm

Input: Grid set *GS*, Density threshold *T*
Output: Intensive grid set *GS′*
1. **Initial** intensive grid set *GS′*
2. **For each** grid *G* in *GS*
3. **If** *density(G)* < *T*
4. **If** G is directly associated with intensive grid
5. Add *G* to *GS′*
6. **Else** let data center be the center of *G* and move grid
7. **If** *density(G)* > *T*
8. *G* is cluster bound grid and add it to *GS′*
9. **Else**
10. *G* is noise grid and abandon it
11. **Else**
12. Add *G* to *GS′*
13. **Return** *GS′*

The grid side length is set to one, and the density threshold is set to three in Figure 3. After the grid cells are divided, it is judged that the grids E and G are dense grid cells according to the principle of closing and opening the grid cells and the density threshold. Since grid cells B, D, F, and H are directly associated with dense grid cells, these non-dense grid cells are set as dense. The grid cells C, J, A, and I are further determined whether they are cluster boundary cells. As shown by the red dotted box in the figure, the center point of the grid cell is moved to the data center point to generate a new grid cell, and the grid density is recalculated. If the new grid cell meets the density threshold, it will be set to a dense grid cell. If the density threshold is not met, the data points in the original grid cells are treated as "noise". The new grid cell obtained by moving the grid cell I is a dense grid cell. After the non-dense grid cells C, J, and A are moved, the new grid cells do not meet the density threshold. Eventually, the data points in these non-dense grid cells will be treated as "noise" data.

4.4. Dense Grid Clustering

After the boundary point judgment is completed, the algorithm in this paper needs to perform grid clustering to form multiple clusters. According to the boundary judgment, we obtain dense cells and non-dense cells; then, we need to cluster all directly related grid cells into a cluster, and the noise points do not participate in the clustering process.

The clustering process uses a depth-first method to find the associated dense grid cells; we combine these associated dense grid cells into the same grid set, and finally map the data points to the corresponding cluster. The dense grid clustering algorithm is described as Algorithm 2.

Algorithm 2 uses the principle of recursion to search all non-clustered grids using depth-first traversal. The main idea of depth first traversal is as follows. Firstly, we take an unreachable vertex as the starting vertex and walk along the edge of the current vertex to the unreachable vertex. Then, when there are no vertices that have not been visited, we return to the previous vertex and continue to explore other vertices until all vertices are visited. In brief, the process of depth first search is to walk along one path to the end, then backtrack, and then do the same walk along another path until all vertices have been visited.

Algorithm 2. Dense grid clustering algorithm(DGCA)

Input: Intensive grid set *GS′*, Intensive grid *G*
Output: Cluster set *CS*
1. **Init** cluster set *CS*
2. **For each** unclustered grid *G′* in *GS′*
3. **If** *G′* is not clustered and *G* is directly associated with *G*
4. Add *G′* to *CS* and remove *G′* in *GS′*
5. DGCA(*GS′*,*G′*)
6. **Return** *CS*

An example of DGCA algorithm is shown in Figure 4. Assume that the traversal starts from the D grid unit, the cluster number of the D grid cell is one, and all the grid cells are judged, E is directly related to it, and the cluster number of the E grid cell is one; then, we continue deep traversal with the E grid cell; at this time, the cluster number to which B belongs is one, the untraversed grid units that are not directly associated with the B grid cell, then we return to cell E, F is directly associated with it, and the cluster number to which it belongs is also one; at this time, the grid cells that are not directly associated with F and have not been traversed then fall back to E. By analogy, all grid cells are judged. There are grid cells G, H, and I in cluster one. The grid cells that complete cluster one are marked with blue grids in the figure. The data points in grid cells A, C and J are "noise" data.

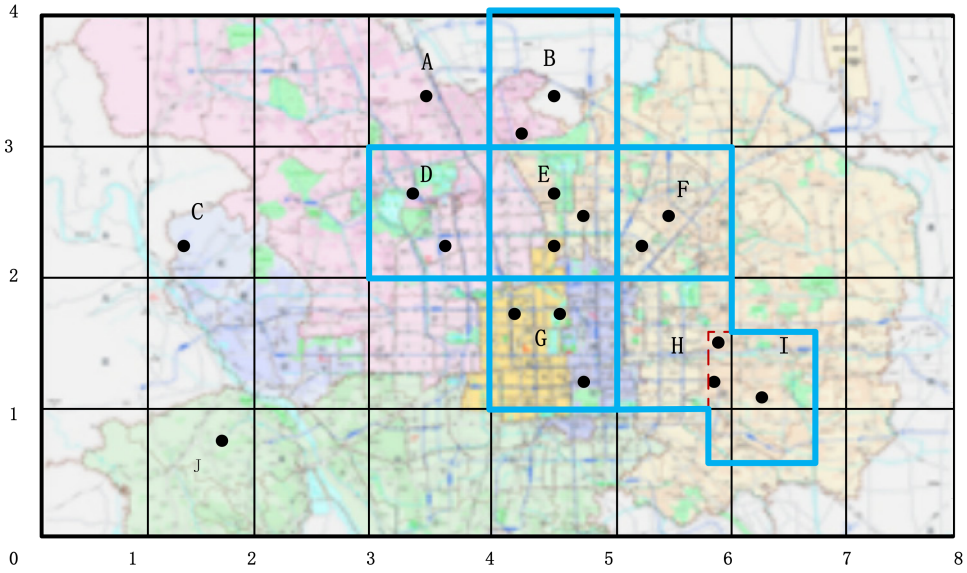


Figure 4. Clustering of dense grid cells.

5. Experimental Results and Analysis

5.1. Datasets and Experimental Environment

The experiment randomly intercepted part of the data from the original T-drive position data set [32,33] to generate four sets of experimental data with different data amounts, as shown in Table 2. The T-Drive trajectory dataset contains one-week trajectories of 10,357 taxis. The total number of points in this dataset is about 15 million and the total distance of the trajectories reaches 9 million kilometers. Among them, DS1 is two sets of taxi position data with numbers 7 and 13; DS2 is four sets of taxi position data with numbers 36, 37, 112, 114; DS3 is nine sets of taxi location data for 427 and 501; DS4 is five sets of taxi location data with numbers 3090, 8249, 9174, 9500, and 9837.

Table 2. Number of sampling points in the dataset.

Dataset	Number of Sampling Points
DS1	822
DS2	1569
DS3	3162
DS4	13,856

The experimental environment is the following: Windows 10 64-bit operating system, Inter Core i5-5350U processor, 8G memory, Visual C # language, based on Microsoft Visual Studio 2015 integrated development environment and SQL Server 2014 database.

5.2. Filter Analysis of Stay Point

Due to positioning errors, the actual data of the stay points may not be completely unchanged; it may change within a small positioning inaccuracy range. This paper defines the stay point threshold to reduce the impact of stay point positioning errors on clustering during the movement of the vehicle; the position of the stay point is allowed to shift within a small range.

In order to analyze the impact of the stay threshold on the experimental data set of this paper, we analyzed the influence of the stay threshold in the pre-processing of stay point filtering of four data sets with different data volumes. According to the stay point events defined in Table 1, the stay time of the experiment in this section is set as follows: DS1 stay time is 15 min, and the remaining data set stay time is 30 min.

The experiment compares the number of samples retained at the point where the thresholds of the original data are 0, 0.0001, 0.001, and 0.01 respectively. The results are shown in Table 3.

Table 3. Retained sampling points after filtering.

Dataset	Stay Point Threshold			
	0	**0.0001**	**0.001**	**0.01**
DS1	695	684	678	566
DS2	1398	1260	1180	918
DS3	2575	2236	2033	1475
DS4	10,194	8384	7323	3377

The purpose of the stay point filtering is intended to delete the stay points considered as noise points. Therefore, Table 3 lists the number of retained points in different data sets when different thresholds are applied. The fewer the retained points, the more stay points.

It is shown in Table 3 that there are a large number of re-sampled data points due to taxi stays in the four data sets. When the stay point threshold is 0, there are some data points that do not move at all in the four data sets. When the stay point threshold is 0.0001, 0.001, and 0.01, the stay point threshold is increased and more stay points are filtered, so the retention points are reduced. The differences between the stay point thresholds at 0, 0.0001,

and 0.001 are not significant, but at 0.01, the data retention is greatly affected, especially in DS3 and DS4. In DS3, the reserved data is reduced from 2575 to 1475. In DS4, the retained data is reduced from 10,194 to 3377. So, if the threshold is too large, the location of normal driving is recognized as the stop, too many points are removed, and it is not suitable to take such a large threshold. Therefore, the threshold in the stay point filtering process needs to be judged and determined according to the actual situation.

According to the classification in Table 1, we analyzed the five stay point events of waiting for traffic lights, getting on and off, traffic jams, business suspension, and business breaks in the experiment in this paper, and we analyzed them on four data sets respectively. When the latitude and longitude are judged, it is judged that the difference between the latitude and longitude is 0. The experimental results are shown in Table 4.

Table 4. Number of stay points resulting from different events.

Dataset	Waiting for Traffic Lights	Getting On and Off	Traffic Jams	Business Suspension	Business Breaks
DS1	52	22	64	11	5
DS2	68	2	101	15	5
DS3	463	21	103	15	10
DS4	2440	550	672	265	365

It can be seen from Table 4 that in the four sets of datasets, taxi stay times of less than 30 min accounted for the majority, which shows that the stay events are mainly caused by waiting for traffic lights, passengers and traffic jams. There are fewer stay points filtered by business suspension and business breaks. From Table 4, it can be seen that the definition of stay point events in this paper considers real scenes. This pre-processing method for stay point filtering is more realistic and accurate.

5.3. Grid Mapping Analysis

We analyzed the effect of the stay point threshold on the density of a single grid cell in the DS3 data set. There were 137 experimental grid cells. The experimental results of grid cells 1–40 are shown in Figure 5. The abscissa is the grid cell number, and the ordinate represents the number of sampling points in each grid cell after mapping, which is the basis for judging whether the grid units are dense.

It is shown in Figure 5 that as the stay point threshold increases, the density of each grid cell shows a decreasing trend when it is 0, 0.0001, 0.001, 0.01. The density is inversely proportional, and the decrease in density indicates the decrease in available data sampling points during the clustering process. When the threshold is 0.01, the density of most grid cells declines faster than 0, 0.0001 and 0.001, and the density of grid cells, as an important measurement index in clustering, is directly related to whether the grid cell is dense. If the density drops too fast, there are not enough grid cells to meet the judgment criteria of dense grids in the subsequent clustering process, so that there are too few available cluster grids, which affects the clustering results. Therefore, the stay point threshold in this paper is not suitable to take larger values such as 0.01.

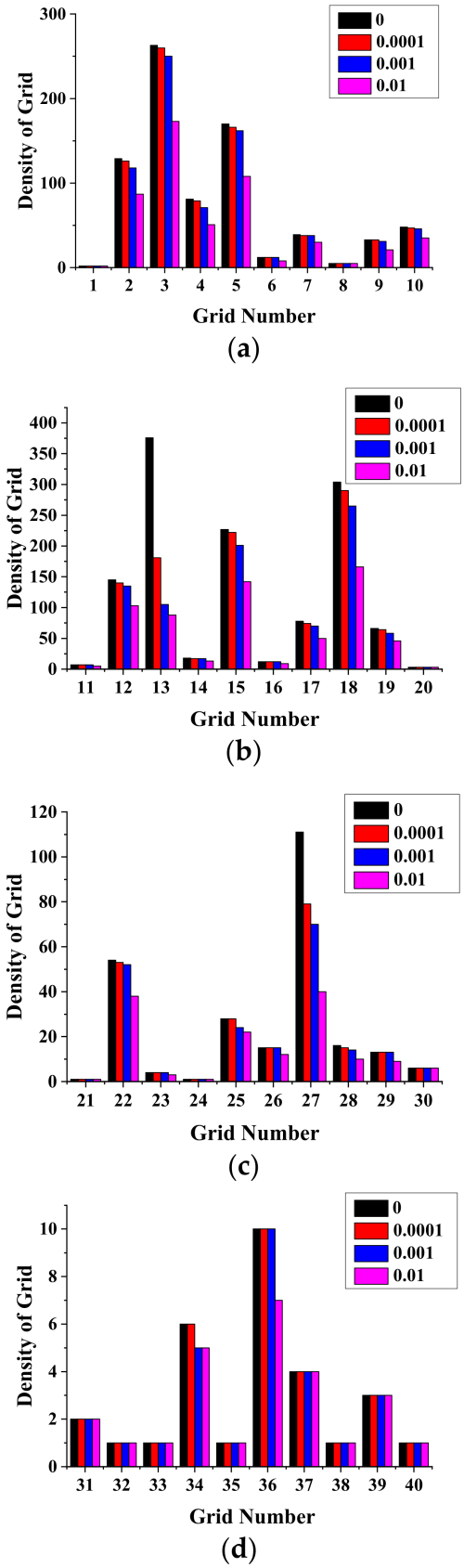


Figure 5. The effect of stay point threshold on grid cell density. Results of (**a**) No. 1–10 grid cells (**b**) No. 11–20 grid cells (**c**) No. 21–30 grid cells (**d**) No. 21–30 grid cells.

5.4. Visual Analysis of Dense Grid Clustering

In this group of experiments, the stay point threshold was set to 0, the DS1 traffic jam residence time Δt was set to within 15 min, and the DS2, DS3, DS4 traffic jam residence time Δt was set to within 30 min in the pre-treatment stage. In order to show the influence of different grid side lengths on the clustering results in the grid mapping stage, the DS1 and DS4 grid side lengths were set to 0.01, and the DS2 and DS3 grid side lengths were set to 0.05. The grid cell density threshold was set to 10; that is, when the number of sampling points in a grid cell was 10, it was determined as a dense grid cell.

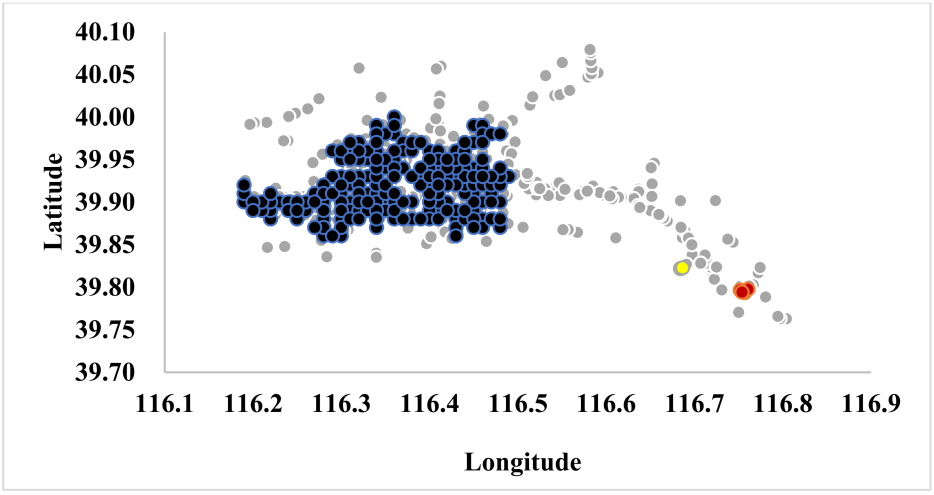
The visualization results of dense grid clustering are shown in Figure 6. Figure 6 is a combination diagram of dense grid cells and the original location point distribution. The figure first shows the clusters composed of dense grid cells in the foreground, and secondly shows the distribution of data sampling points in the form of background color. The data sampling points of dense grid cells are the data points in the cluster, and the sampling points that do not exist in any grid cell are sparse noise points.

The light gray points in Figure 6 are the data sampling points, that is, the distribution of the position points in different data sets after filtering the stay points. The dark points such as blue, red, and yellow represent the cluster grid points. Since the side length of the experimental grid cell has been given, the grid cell can be uniquely determined according to any grid endpoint. Therefore, the grid cell is represented by the endpoint at the lower left of the grid cell in order to simplify the graphic display.

Figure 6 shows that the proposed clustering can effectively determine sparse noise points. For example, there are a large number of sparse points at 115.5–116.1 degrees north latitude and 39.7–40.05 degrees east longitude in the DS2 result of Figure 6b, which are not included in any clusters. Moreover, the experimental results in other data sets also show that similar sparse points have no effect on the clustering results, for example, the sampling points around the two clusters and at the joint in DS1 clustering results in Figure 6a.

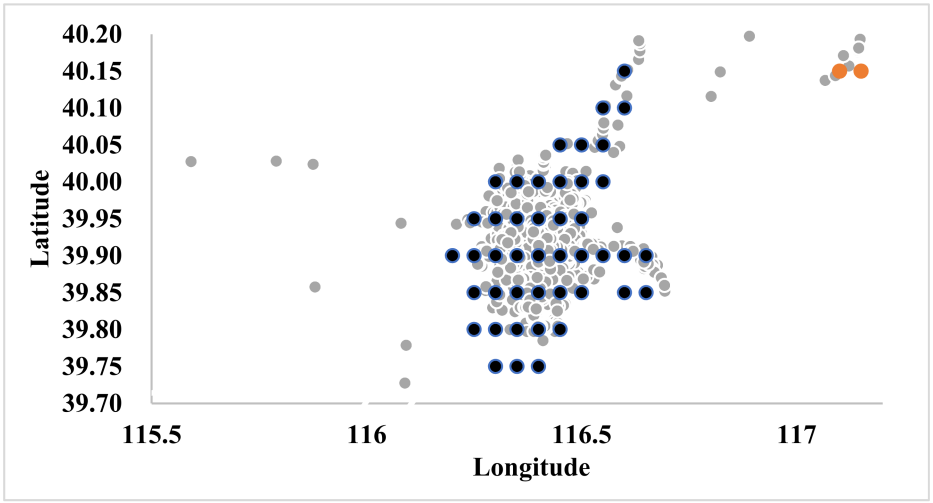
The four sets of experimental results in Figure 6a–d show that the method in this paper can accurately determine the clustering of the high-density areas of the sampling points, which represents the high-density areas of the taxi distribution, and it has a good effect on the extraction of urban hotspots.

Figure 6 also shows the influence of grid cell side length on grid mapping. In Figure 6a,d, the grid cells of DS1 and DS4 are denser. In Figure 6b,c, the grid cells of DS2 and DS3 are relatively sparse. This is because the side length of the grid of DS1 and DS4 is set to 0.01 and that of DS2 and DS3 is set to 0.05. The different length of the grid leads to different density and sparseness, which is caused by these four sets of data being collected from Beijing taxis, so the spatial range of the data is not large.

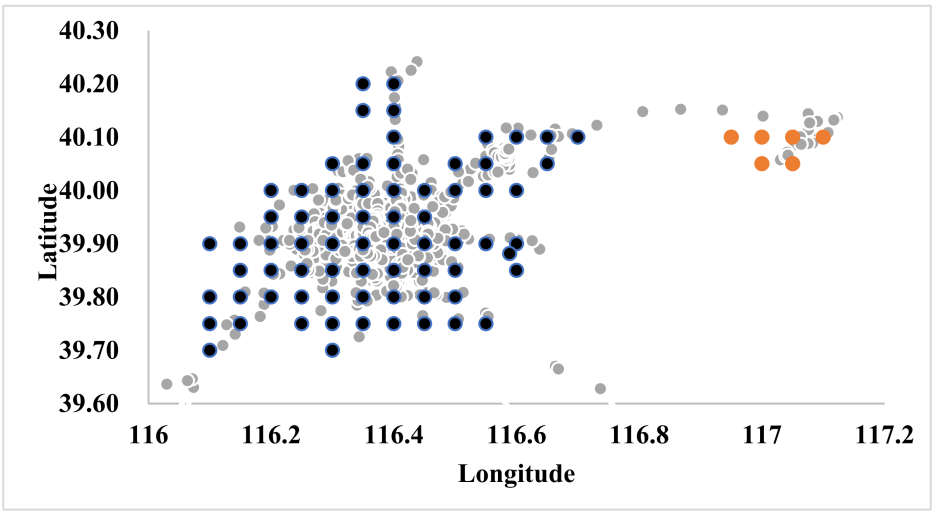


(**a**)

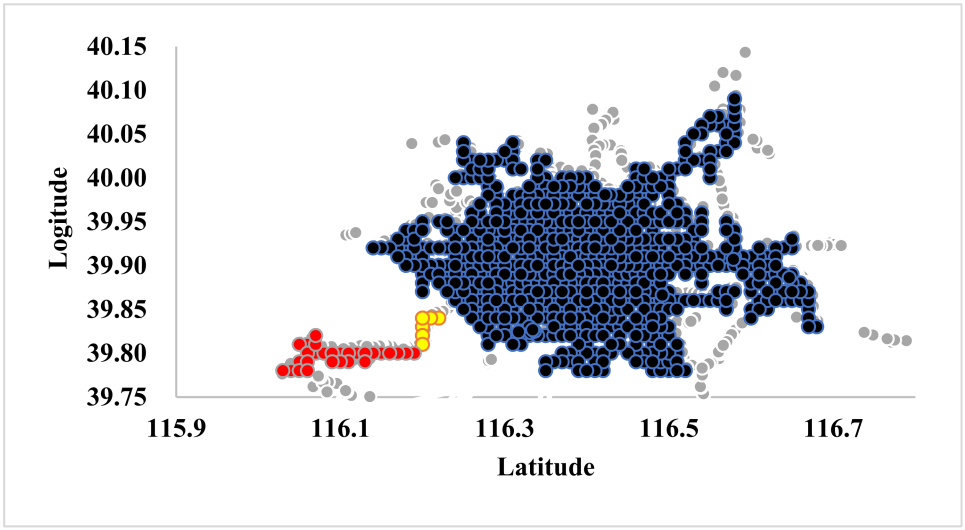
Figure 6. *Cont.*



(b)



(c)



(d)

Figure 6. Visualization of clustering results on different data sets. Clustering results on (**a**) DS1 (**b**) DS2 (**c**) DS3 (**d**) DS4.

5.5. Analysis of Comparative Experiment Results

In this paper, the clustering method-based on stay point and grid density(CMSPGD) and Hybrid Feature-based DBSCAN(HF_DBSCAN) [30], Effective Parameter Selection Process for the DBSCAN(PS_DBSCAN) [34] were compared.

(1) HF_DBSCAN

HF_DBSCAN is an algorithm based on improved DBSCAN proposed by Luo et al. in 2017 [30]. DBSCAN is a classic density-based algorithm used to find high-density areas in space, and different derivatives of the algorithm have been proposed to find urban hotspot areas. The density of the current point in the DBSCAN algorithm is determined by the distance from the current point. The number of points within a certain distance is used for balance. The HF_DBSCAN algorithm uses a Gaussian function as the density of points. The calculation method is as Formula (8).

$$\varphi(p_i) = \sum_{j=1}^{n} e^{-\left(\frac{d_{ij}}{\sigma_1}\right)^2} \tag{8}$$

where $p_i(i = 1, 2, 3 \ldots, n)$ represents the point, d_{ij} represents the Euclidean distance between p_i and p_j, and σ_1 represents the standard deviation. The standard deviation in this experiment is 0.3.

(2) PS_DBSCAN

PS_DBSCAN is an improved algorithm proposed by Huang et al. in ACM Trans in 2019 [34]. For the original DBSCAN algorithm, there is no strict index determination for the selection of two parameters of radius length and density threshold, resulting in inaccurate clustering. The author improved the method for determining these two sets of parameters with the following steps. Firstly, the author determined a larger radius length, and then gradually reduced the radius length. The author observed the number of clusters for each radius length cluster density threshold comparison; as a result, the author found the density threshold when the number of clusters just decreased as the density threshold rose, and set it to the appropriate density threshold under the radius length of the group. The density threshold of the last set of the above changes is the final value. The author observed the comparison between the number of clusters and the radius length under the appropriate density threshold obtained in the previous step. The radius length corresponding to the larger number of clusters is the appropriate value.

In this paper, DS4 is first tested according to the parameter selection method in the PS_DBSCAN algorithm to find the appropriate radius length and density threshold. First, we determined a larger radius length of 0.025, and then reduced it to 0.01 and 0.005 in sequence. The comparison results of the density threshold and the number of clusters under these three groups of radius lengths are shown in Tables 5–7 below.

Table 5. Radius length = 0.005.

Density Threshold	Number of Clusters
50	10
60	13
70	10
80	6
100	4
150	3

Table 6. Radius length = 0.01.

Density Threshold	Number of Clusters
10	5
20	8
25	8
100	8
110	9
120	8
150	4

Table 7. Radius length = 0.025.

Density Threshold	Number of Clusters
60	5
80	5
100	6
150	7
200	6
250	3
300	2

First of all, it is judged that there are three groups in which the density threshold increases and the number of clusters decreases in the three sets of data: radius length 0.005 and density threshold 60, radius length 0.01 and density threshold 110, and radius length 0.025 and density threshold 150. Among these three sets of data, the density threshold 150 is the largest and is the key value for the last change, so it is used as a suitable density threshold parameter. Then, among the three groups of data, the data with density threshold of 150 is as follows: the density threshold with a radius length of 0.025 is 150 and the number of clusters is three; the density threshold is 150 with a radius length of 0.005 and the number of clusters is three; the radius length is 0.025, the density threshold is 150, and the number of clusters is seven. Therefore, for DS4, the appropriate radius length of the DBSCAN algorithm based on parameter selection is 0.025, and the density threshold is 150.

Similarly, the appropriate radius lengths for DS1, DS2, and DS3 are 0.005, 0.025, and 0.025, respectively, and the density thresholds are 10, 30, and 50, respectively.

(3) Contrast analysis of clustering accuracy

In this paper, the experimental clustering results of HF_DBSCAN and PS_DBSCAN in four data sets are shown in Tables 8–19.

Table 8. The clustering results of the algorithm in this paper for the DS1 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	114	116.3474	39.91577	0.727285	0.006397
2	22	116.7604	39.79758	0.068458	0.003112
3	12	116.691	39.82763	0.019308	0.001908

Table 9. The clustering results of HF_DBSCAN for the DS1 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	517	116.387	39.88598	31.92563	0.061752
2	2	116.5571	39.86765	0.00608	0.00304
3	2	116.3395	39.83978	0.000322	0.000161
4	1	116.3452	40.02308	0	0
5	3	116.4102	40.05595	0.005303	0.001768
6	2	116.5137	40.0139	0.003266	0.001633

Table 10. The clustering results of PS_DBSCAN for the DS1 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	14	116.7604	39.79758	0.021774	0.001555
2	1	116.6914	39.8281	0	0
3	66	116.3535	39.91067	0.199025	0.003016

Table 11. The clustering results of the algorithm in this paper for the DS2 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	1299	116.4454	39.79803	132.1319	0.101718
2	30	117.1443	40.18103	0.073971	0.002466

Table 12. The clustering results of HF_DBSCAN for the DS2 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	1020	116.4454	39.79803	66.49093	0.065187
2	1	116.5831	39.93808	0	0
3	2	116.4669	40.05162	0	0

Table 13. The clustering results of PS_DBSCAN for the DS2 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	861	116.4115	39.80142	44.88626	0.052133
2	199	116.6954	39.85161	0.09817	0.000493
3	7	116.6387	39.89374	0.044806	0.006401

Table 14. The clustering results of the algorithm in this paper for the DS3 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	2351	116.3792	39.92107	183.9485	0.078243
2	114	117.0397	40.06385	2.284611	0.02004

Table 15. The clustering results of HF_DBSCAN for the DS3 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	2182	116.3682	39.87703	1	0.05068
2	1	116.4473	39.7648	2	0
3	1	116.2813	39.7741	3	0
4	1	116.4486	40.05765	4	0
5	1	116.2374	40.00023	5	0
6	1	116.2137	39.97259	6	0
7	1	116.202	39.93237	7	0
8	2	116.5125	39.80896	8	0.00023
9	1	116.5558	39.90857	9	0

Table 16. The clustering results of PS_DBSCAN for the DS3 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	1835	116.3716	39.91364	104.6616	0.057036
2	107	116.5826	40.06608	0.900239	0.008413
3	76	117.0397	40.06383	0.149771	0.001971

Table 17. The clustering results of the algorithm in this paper for the DS4 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	9316	116.4214	39.90342	1045.187	0.112193
2	15	116.2075	39.82121	0.059564	0.003971
3	261	116.0665	39.81623	7.000711	0.026823

Table 18. The clustering results of HF_DBSCAN for the DS4 dataset.

No	m	Longitude	Latitude	LoadLength	Avg
1	10117	116.4167	39.90212	1246.077	0.123167
2	2	116.1866	40.03918	0.00035	0.000175
3	4	116.1074	39.93499	0.017727	0.004432
4	7	116.7657	39.81631	0.095028	0.013575
5	17	116.8653	39.80081	0.5669	0.033347
6	3	116.606	40.1512	0.018472	0.006157
7	8	116.6355	40.16783	0.044321	0.00554
8	4	116.6416	40.22429	0.010032	0.002508
9	4	116.6335	40.27486	0.003967	0.000992
10	3	116.6492	40.31469	0.004351	0.00145
11	1	116.1681	39.65133	0	0

Table 19. The clustering results of PS_DBSCAN for the DS1 dataset.

No	M	Longitude	Latitude	LoadLength	Avg
1	6285	116.40066	39.90026	426.1125402	0.067798336
2	162	116.06531	39.81733	0.752344892	0.004644104
3	239	116.58334	40.05039	1.707618382	0.007144847
4	232	116.62613	39.90084	0.559221779	0.002410439
5	246	116.19529	39.91437	2.731134047	0.011102171
6	1175	116.67104	39.84935	0.792695798	0.000674635
7	177	116.48852	39.78185	1.180586605	0.006669981

The attribute *No* in the table indicates the number of the cluster, *m* indicates the number of data points in the cluster; the larger *m*, the more points participating in the cluster, and the fewer noise points discarded. *Longitude* and *Latitude* are the cluster center coordinates of the cluster, that is, the distance and minimum of all points in the cluster to the point; *LoadLength* represents the clustering distance of the cluster, the sum of the distances from all points to the cluster center. The calculation is as Formula (9).

$$LoadLength = \sum_{p \in cluster} dis(p, center) \tag{9}$$

where *p* represents the cluster element in the cluster; *center* represents the clustering center of the cluster, namely Longitude and Latitude coordinates; *Avg* represents the average aggregation distance of each point, calculated as in Formula (10).

$$Avg = \frac{LoadLength}{m} \tag{10}$$

Avg represents the average density of points in the cluster. The greater the *Avg*, the denser the points in the cluster. If there are more points in the cluster and the denser, the better the clustering effect of each cluster.

Tables 8–10 show that the proposed algorithm and PS_DBSCAN algorithm have fewer *m* values than the HF_DBSCAN algorithm for the DS1 dataset, indicating that in small-scale data sets, the algorithm and PS_DBSCAN algorithm will have more cluster sampling points lost. Secondly, the values of *LoadLength* and *Avg* in the group table show that the distance within the cluster generated by the HF_DBSCAN algorithm is large, indicating

that the clustering accuracy quality is not as good as the algorithm of this paper and the PS_DBSCAN algorithm.

Tables 11–19 show that the m values of the three algorithms in the DS2, DS3, and DS4 data sets are not much different, indicating that the three sets of algorithms are basically the same in the number of sampling points of the clustering results. The comparison algorithm is higher, which shows that the clustering accuracy of the algorithm in this paper is worse in the intra-cluster distance. This is because the grid mapping process of the clustering in this paper will bring a certain accuracy loss.

Tables 8–19 show that although the HF_DBSCAN algorithm produces more clusters, most of the clusters have sparse points and fewer elements. The clustering algorithm and the PS_DBSCAN algorithm in this paper result in more uniform data. For example, Tables 14–16 shows that the clustering method in this paper and PS_DBSCAN algorithm generate two to three clusters and the HF_DBSCAN algorithm generates nine clusters, but according to the value of m, the clusters 2, 3, 4, 5, 6, 7, and 9 have only one data point. This type of experimental data can be removed as noise or merged into other clusters. Similar results have been obtained for other datasets. The clusters formed by the clustering algorithm in this paper and PS_DBSCAN algorithm are more reasonable, balanced and stable. However, in the PS_DBSCAN algorithm, the parameters are optimized to make the clustering results more uniform, and the algorithm is relatively complicated to implement. Therefore, this paper's algorithm is simpler and more efficient than the PS_DBSCAN algorithm to implement in the formation of reasonable clusters.

In summary, compared to the PS_DBSCAN algorithm in terms of clustering effect, the algorithm in this paper is simpler and discards fewer noise points. Compared to the HF_DBSCAN algorithm, the clusters formed by this method are more uniform and reasonable.

(4) Comparative analysis of running time

The experiment also compares and analyzes the execution time of the algorithm in this paper with the HF_DBSCA N and PS_DBSCAN algorithms. The experimental results for four data sets are shown in Table 20.

Table 20. Algorithm execution time comparison(s).

DataSet	CMSPGD	HF_DBSCAN	PS_DBSCAN
DS1	0.088	0.172	0.34
DS2	0.102	0.693	0.764
DS3	0.382	2.235	2.401
DS4	2.126	36.534	35.173

Table 20 shows that the running time consumption of the clustering algorithm in this paper when processing the same size data set is much lower than that of the comparison algorithm. As the number of data object sets continues to increase, the running time of the comparison algorithm increases sharply. In this paper, the increase in the running time of the grid-based and density clustering algorithm based on grid cells is much smaller than that of the comparison algorithm. It has advantages over comparison algorithms when dealing with large datasets. This is because the algorithm uses a grid clustering algorithm to divide the grid, so that the processed object is not a data point, but a divided grid cell, and the improved DBSCAN clustering algorithm operates on data objects, so our clustering algorithm is more efficient than the HF_DBSCAN and PS_DBSCAN algorithms.

In this experiment, clustering is performed for grid cells. The number of grid cells after space division and the number of non-dense cells will also affect the efficiency of this experiment, and further judgments on non-dense grid cells are also required. But the overall efficiency is still significantly better than the comparison algorithm.

6. Conclusions

In this paper, a clustering method based on stay points and grid density is proposed. First, the stay point filtering algorithm is used to avoid the impact of taxi stop events on the density of grid cells. Secondly, grid mapping and cluster boundary point judgment are used to avoid the influence of sparse noise points on the sampling set while performing cluster mining. Finally, the grid clustering method is used to significantly improve the time efficiency of the existing methods. However, the classification of stay points is mainly from the perspective of stay time. The granularity is not further refined, which may cause misjudgment of some stay events. In the future work, the realistic semantic basis for judgment of the stay point will be enhanced and the stay point category will be refined, making the research on stay point filtering more in-depth. Furthermore, the criteria for judging whether two cell grids are in the same cluster mainly considering the number of sampling points is not comprehensive enough. In the clustering process, a further judgment criterion is whether the sampling point distribution in the two grid units is concentrated, which can lead to more accurate grid clustering.

Author Contributions: Conceptualization, Xiaohan Wang; methodology, Zepei Zhang and Yonglong Luo; software, Zepei Zhang; validation, Xiaohan Wang and Zepei Zhang; writing—original draft preparation, Xiaohan Wang and Zepei Zhang; writing—review and editing, Yonglong Luo; visualization, Zepei Zhang; supervision, Yonglong Luo; project administration, Yonglong Luo; funding acquisition, Yonglong Luo and Xiaohan Wang. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 61972439) and the Natural Science Foundation of Anhui Province (Grant No. 2008085MF212).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Scholz, R.W.; Lu, Y. Detection of dynamic activity patterns at a collective level from large-volume trajectory data. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 946–963. [CrossRef]
2. Shan, J.; Alves, A.; Rodrigues, F.; Ferreira, J., Jr.; Pereira, F.C. Mining point-of-interest data from social networks for urban land use classification and disaggregation. *Comput. Environ. Urban Syst.* **2015**, *53*, 36–46.
3. Yue, Y.; Zhuang, Y.; Yeh, A.G.O.; Xie, J.Y.; Ma, C.L.; Li, Q.Q. Measurements of POI-based mixed use and their relationships with neighbourhood vibrancy. *Int. J. Geogr. Inf. Syst.* **2017**, *31*, 658–675. [CrossRef]
4. Ming, N.; He, Q.; Jing, G. Forecasting the Subway Passenger Flow Under Event Occurrences With Social Media. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1623–1632.
5. Yang, X.; Chen, A.; Ning, B.; Tang, T. Measuring Route Diversity for Urban Rail Transit Networks: A Case Study of the Beijing Metro Network. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 259–268. [CrossRef]
6. Zhang, F.; Zhao, J.; Tian, C.; Xu, C.; Liu, X.; Rao, L. Spatio-Temporal Segmentation of Metro Trips Using Smart Card Data. *IEEE Trans. Veh. Technol.* **2016**, *65*, 1137–1149. [CrossRef]
7. Le, M.K.; Bhaskar, A.; Chung, E. Passenger Segmentation Using Smart Card Data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1537–1548.
8. Itoh, M.; Yokoyama, D.; Toyoda, M.; Tomita, Y.; Kawamura, S.; Kitsuregawa, M. Visual Exploration of Changes in Passenger Flows and Tweets on Mega-City Metro Network. *IEEE Trans. Big Data* **2016**, *2*, 85–99. [CrossRef]
9. Chen, Y.; Chao, S. Performance Analysis of Smartphone-Sensor Behavior for Human Activity Recognition. *IEEE Access* **2017**, *5*, 3095–3110. [CrossRef]
10. Zhang, W.; Qi, G.; Gang, P.; Hua, L.; Li, S.; Wu, Z. City-Scale Social Event Detection and Evaluation with Taxi Traces. *Acm Trans. Intell. Syst. Technol.* **2015**, *6*, 1–20. [CrossRef]
11. Fuchs, G.; Stange, H.; Hecker, D.; Andrienko, N.; Andrienko, G. Constructing semantic interpretation of routine and anomalous mobility behaviors from big data. *Sigspatial Spec.* **2015**, *7*, 27–34. [CrossRef]
12. Unsalan, C. Measuring Land Development in Urban Regions Using Graph Theoretical and Conditional Statistical Features. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3989–3999. [CrossRef]
13. Yin, J.; Soliman, A.; Yin, D.; Wang, S. Depicting urban boundaries from a mobility network of spatial interactions: A case study of Great Britain with geo-located Twitter data. *Int. J. Geogr. Inf. Syst.* **2017**, *31*, 1293–1313. [CrossRef]

14. Yuan, N.J.; Zheng, Y.; Xie, X.; Wang, Y.; Xiong, H. Discovering Urban Functional Zones Using Latent Activity Trajectories. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 712–725. [CrossRef]
15. Sarkar, S.; Chawla, S.; Parambath, S.P.; Srivastava, J.; Borge-Holthoefer, J. Effective Urban Structure Inference from Traffic Flow Dynamics. *IEEE Trans. Big Data* **2017**, *3*, 181–193. [CrossRef]
16. Kong, X.; Feng, X.; Wang, J.; Rahim, A.; Das, S.K. Time-Location-Relationship Combined Service Recommendation Based on Taxi Trajectory Data. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1202–1212. [CrossRef]
17. Lee, J.; Shin, I.; Park, G.-L. Analysis of the Passenger Pick-Up Pattern for Taxi Location Recommendation. In Proceedings of the Fourth International Conference on Networked Computing and Advanced Information Management, Gyeongju, Korea, 2–4 September 2008; Volume 1, pp. 199–204.
18. Thuillier, E.; Moalic, L.; Lamrous, S.; Caminada, A. Clustering Weekly Patterns of Human Mobility Through Mobile Phone Data. *IEEE Trans. Mob. Comput.* **2018**, *17*, 817–830. [CrossRef]
19. Jain, A.K.; Law, M.H.C. Data clustering: A user's dilemma. In *International Conference on Pattern Recognition and Machine Intelligence*; PReMI 2005. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3776, pp. 1–10. [CrossRef]
20. Shi, X.; Guo, Z.; Xing, F.; Cai, J.; Yang, L. Self-learning for face clustering. *Pattern Recognit.* **2018**, *79*, 279–289. [CrossRef]
21. Diego, I.M.d.; Siordia, O.S.; Fernández-Isabel, A.; Conde, C.; Cabello, E. Subjective data arrangement using clustering techniques for training expert systems. *Expert Syst. Appl.* **2019**, *115*, 1–15. [CrossRef]
22. Mousavirad, S.J.; Ebrahimpour-Komleh, H.; Schaefer, G. Effective image clustering based on human mental search. *Appl. Soft Comput.* **2019**, *78*, 209–220. [CrossRef]
23. Hou, J.; Gao, H.; Li, X. DSets-DBSCAN: A Parameter-Free Clustering Algorithm. *IEEE Trans. Image Process.* **2016**, *25*, 32–39. [CrossRef] [PubMed]
24. Ghaffari, R.; Golpardaz, M.; Helfroush, M.S.; Danyali, H. A fast, weighted CRF algorithm based on a two-step superpixel generation for SAR image segmentation. *Int. J. Remote Sens.* **2020**, *41*, 3535–3557. [CrossRef]
25. Zhou, T.; Liu, X.; Qian, Z.; Chen, H.; Tao, F. Dynamic Update and Monitoring of AOI Entrance via Spatiotemporal Clustering of Drop-Off Points. *Sustainability* **2019**, *11*, 6870. [CrossRef]
26. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [CrossRef] [PubMed]
27. Schoier, G.; Borruso, G. Individual movements and geographical data mining clustering algorithms for highlighting hotspots in personal navigation routes. In Proceedings of the 2011 International Conference on Computational Science, Santander, Spain, 20–23 June 2011; pp. 454–465.
28. Zhou, C.; Dan, F.; Ludford, P.; Shekhar, S.; Terveen, L. Discovering personally meaningful places:An interactive clustering approach. *ACM Trans. Inf. Syst.* **2007**, *25*, 12–17. [CrossRef]
29. Hwang, S.; Evans, C.; Hanke, T. Detecting Stop Episodes from GPS Trajectories with Gaps. In *Seeing Cities through Big Data*; Springer: Cham, Switzerland, 2017; pp. 427–439.
30. 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*, 63–74. [CrossRef]
31. Zhao, Q.; Shi, Y.; Liu, Q.; Franti, P. A grid-growing clustering algorithm for geo-spatial data. *Pattern Recognit. Lett.* **2015**, *53*, 77–84. [CrossRef]
32. Yuan, J.; Zheng, Y.; Xie, X.; Sun, G. Driving with knowledge from the physical world. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 316–324.
33. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; Huang, Y. T-drive: Driving directions based on taxi trajectories. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 99–108.
34. Huang, Y.; Xiao, Z.; Yu, X.; Wang, D.; Havyarimana, V.; Bai, J. Road Network Construction with Complex Intersections Based on Sparsely Sampled Private Car Trajectory Data. *ACM Trans. Knowl. Discov. Data* **2019**, *13*, 35. [CrossRef]