Traffic travel pattern recognition based on sparse Global Positioning System trajectory data

Juan Chen, Kepei Qi and Shiyu Zhu

Abstract
This article mainly uses sparse Global Positioning System trajectory data to identify traffic travel pattern. In this article, the data are preprocessed and the eigenvalues are calculated. Then, the Global Positioning System track points are identified and extracted by walking and non-walking segments. Finally, the three machine learning models of support-vector machine, decision tree, and convolutional neural network are used for comparison experiments. The innovation of this article is to propose a walking and non-walking identification method based on density-based spatial clustering of applications with noise clustering. The method takes into account the continuous state between the geographical distributions, and it has better noise immunity against the influence of external factors. In this process, this article directly achieves better conversion point recognition results through the Global Positioning System track point information, which lays a good foundation for the accuracy of travel pattern recognition. The experimental results of this article show that compared with threshold-based and multi-layer perceptron–based methods, the recognition method based on density-based spatial clustering of applications with noise clustering has the highest accuracy, reaching 82.20%. Then, support-vector machine, decision tree, and convolutional neural network are used for traffic travel pattern recognition. The F1-score of support-vector machine is the highest, reaching 0.84, and the F1-scores of decision tree and convolutional neural network are 0.78 and 0.80, respectively. Finally, the support-vector machine was used for the recognition test to achieve an accuracy of 76.83%.

Keywords
Traffic travel pattern, trajectory segmentation, transition point, density-based spatial clustering of applications with noise clustering, convolution neural network

Date received: 17 October 2019; accepted: 2 October 2020

Handling Editor: Lyudmila Mihaylova

Introduction
Behind the rapid development of urban transportation network, there is a problem of the shortage of transportation resources caused by the urban area and population surge. How to make urban transportation modernization and intelligent development become the focus of the government, citizens, and enterprises. With the revolutionary upgrading of information technology, people have generated a large amount of travel information in their lives, and these information are collected as data, integrated, processed, and analyzed, providing new solutions to solve urban traffic problems, which makes the application of big data technology in traffic field received extensive attention. By processing and analyzing a series of Global Positioning System (GPS) travel data, digital map service providers...
can help users to provide the best travel plan, and the route recommendation scheme combined with real-time road network traffic conditions can effectively alleviate traffic congestion, morning and evening peaks, and other city traffic problems. Therefore, the travel pattern recognition problem based on GPS trajectory data can benefit multiple parties. However, due to the fact that travel data originate from real life, the accuracy of GPS equipment, the sparseness of data, the differences of travel individuals, and the alternation of multiple travel patterns lead to the complexity of the problem, which makes the problem of far-reaching theoretical research.

For the problem of travel pattern recognition, quite a few scholars have conducted in-depth research on it. Some researchers have used the idea of segmentation between walking and non-walking segments based on transition points, and used conditional random field (CRF) to classify GPS data for travel patterns. Some researchers have analyzed the shortcomings of speed-based single feature classification, and introduced three new eigenvalues of speed change rate, parking rate and direction change rate, and combined with traditional features such as interval maximum speed and interval maximum acceleration to classify travel patterns. Some researchers used the convolutional neural network (CNN) in deep learning and Bayesian network for travel pattern classification. All of the above studies have focused on the improvement of the relevant model of the final travel pattern classification, but there is not much in-depth discussion of the previous trajectory segmentation work.

Combined with the research ideas and deficiencies of the above scholars, the research work of this article is as follows:

- First, pre-process GPS trajectory data, including data set field filtering and generation, deduplication, tag correspondence, and data filtering. The base feature values are then calculated based on the unique latitude and longitude of the data set and the time stamp information.
- Second, the trajectory segmentation is performed based on the conversion point recognition technique. First, the walking and non-walking points are identified. The threshold method, multi-layer perceptron, and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering are used to compare the performance. Second, the rules to extract and merge the walking and non-walking segments are set, and the GPS track points at the two transitions are marked as the transition points. Finally, the mark transition points are compared with the real transition points, and the recognition result is evaluated by the recall rate and the precision rate.
- Third, identify different travel patterns. First, the differences between different travel patterns are analyzed in terms of distance, speed, and acceleration to screen the effective eigenvalues. Second, the eigenvalues are calculated and a series of machine learning models are built to classify the travel patterns. The performance of each model and experiment results are analyzed in the end.

The overall research content of the article is shown in Figure 1. The main difficulties of this research are as follows:

- First, the GPS trajectory data collected by the GPS device exist the problem of inaccurate or difficult to correct. Denoising by the corresponding technology will introduce new noise and erase the original trajectory data characteristics. Moreover, the amount of available data in the
data set is not large, and there are many walking data in the trajectory data that occur in the community, the park, or even the building, which is difficult to correct.

- Second, the trajectory data eigenvalues are calculated based on the Haversine formula in geography to calculate the distance between two given points, where radius is one of the important parameters of the formula. However, the radius of the earth is different at different latitudes, and the calculation of the error will have a great impact on the experiment.

- Third, the identification of the transition point is the core of the travel segment division, and the transition point is usually accompanied by a process in which the speed is first decelerated to near zero and then slowly accelerated. This process is very similar to the characteristics of the vehicle waiting for the red light, the arrival and departure of the bus, and the traffic congestion, which is difficult to distinguish correctly.

- Fourth, the characteristics of some travel patterns are relatively close. For example, during the period of traffic congestion, the speed of taxi/private car is very close to the speed of bicycles and even walking, and it is difficult to distinguish accurately.

For the above technical difficulties, this article proposes a new solution for the third problem: this article proposes a walking and non-walking recognition method based on DBSCAN clustering. Since the GPS trajectory points in the data set are mostly collected at a fixed frequency, the walking points are denser than the non-walking points, and the DBSCAN clustering considers the continuous state between the geographical distributions, bringing better noise immunity. In the process of identification, this article directly achieves better transition point recognition results through the GPS track point information, which lays a good foundation for the accuracy of travel pattern recognition.

**Literature reviews**

Travel pattern recognition was first proposed in the United States since the 1960s. In the early days of the study, questionnaires and telephone interviews were the main methods of data collection. Around 2000, the scheme of replacing traditional travel logs with GPS technology to collect travel data was first proposed. Since then, many foreign scholars have proposed characteristic variables such as average speed, maximum speed, maximum acceleration, and travel distance, supplemented by bus route network and site distribution information to identify the travel pattern. With the continuous maturity and application of machine learning, scholars pay more attention to mining GPS data features instead of geographic information system (GIS) information. Models such as decision trees, neural networks, support-vector machines (SVMs), and Bayesian networks have become mainstream algorithm models, and their classification accuracy has been proven to be superior.

Travel data can be divided into two types: one is trajectory data based on a single travel pattern and the other is trajectory data mixed by multiple travel patterns. Because the mixed trajectory data contain a variety of traffic characteristics and the transition and stay between different travel patterns, its complexity and variability increase the difficulty of travel pattern recognition. Therefore, the simplification, reconstruction, and semantic mining of trajectory data become the premise and foundation of solving the problem.

For mixed trajectory data, the travel pattern recognition problem is mainly divided into two parts: one is the trajectory segmentation and the other is the pattern recognition. Considering that the state of a single-track point is highly susceptible to the external environment (e.g., traffic congestion, weather, and individual speed differences), the probability of directly identifying a single-track point is higher. Therefore, the longer the trajectory, the more abundant and accurate travel characteristics of the data. In addition, trajectory data derived from real life often contain multiple travel patterns. Therefore, the primary task of trajectory segmentation is to divide the trajectory into sub-sections with uniform travel patterns as accurately as possible, and the number of sub-sections is as small as possible, travel distance and duration are as long as possible. The second step is to perform pattern recognition on the divided sub-sections, mainly by selecting appropriate feature values and using machine learning models for recognition. This article will review four trajectory segmentation methods and pattern recognition problems.

**Trajectory segmentation**

The core idea of trajectory segmentation is to compute the characteristics of each individual point in the trajectory and to group points with similar features. At present, the commonly used methods for trajectory segmentation are based on unified interval, based on static point, based on GIS data of road network, and based on transition points.

**Trajectory segmentation based on unified interval.** The unified interval method mostly uses a unified time length, a uniform distance, and a unified GPS track point as a segmentation method. Chang and Newman compared these three trajectory segmentation methods
based on uniform interval, and concluded that the performance of these three methods is different in different modes. For example, the unified distance method is more accurate in identifying the taxi/private car mode, and the unified time method is more accurate in recognizing the walking mode. Some researchers also use a fixed period of time as the smallest unit to directly identify the travel pattern of the smallest unit, but after identification, they merge the sections with the same recognition result to improve the accuracy of the segmentation.\textsuperscript{11} However, Zheng et al.\textsuperscript{12} compared the performance differences between two unified interval segmentation methods (uniform duration and uniform distance) and transformation point recognition segmentation methods. They chose the optimal result and transformation point of uniform time length and uniform distance parameter. Compared with the recognition segmentation method, the results show that the unified interval method is inferior to the conversion point recognition segmentation method in almost all evaluation indicators. The main reason is that the unified interval method has a higher probability of including two travel patterns, so it will bring a larger error.

In summary, the shortcomings of the trajectory segmentation method based on unified interval are mainly in the following three aspects. First, the interval selection method of the unified interval method greatly affects the effect of the final trajectory segmentation, its performance is different for different travel modes, and the promotion is poor. Second, the unified interval method needs to select the optimal interval threshold for testing, and the calculation amount is large, and the addition of new data will lead to the change of the optimal threshold and the low adaptability to new data. Third, the unified interval method has a large probability of mixing the GPS track points of multiple travel modes in one interval, so the calculation result of the entire segment feature value does not accurately represent the true characteristics of the travel pattern, and finally leads to the low recognition accuracy.

**Trajectory segmentation based on static point.** By identifying the point where the speed is significantly smaller than the walking speed, the static point recognition method classifies the point into a non-traveling state, while the non-traveling state has a higher probability that the segment has two independent trajectory segments.\textsuperscript{13} Shen\textsuperscript{14} used the threshold method to treat a series of speeds as 0 m/s points as non-travel sections, the first point of the section as the end point of the previous trip section, and the last point as the starting point of next trip section. Some research work uses the density-based static point recognition method. When the pause occurs, the GPS trajectory points will be densely located in the same area, thus completing the trajectory segmentation.\textsuperscript{15,16} Huang et al.\textsuperscript{17} further analyzed the commonality of multiple trajectory data staying regions to analyze the repeated behavior of individual travel. Zhou and Yang\textsuperscript{18} proposed a new space-time clustering algorithm Adjacent Temporospatial (AT)-DBSCAN to identify the stopping point in the trajectory. The algorithm has a stronger generalization ability for complex problems such as indoor activities and positioning drift.

The activity recognition method is similar to the static point recognition. It is to distinguish the activity and the travel status by connecting the last record point of activity i with the first record point of activity i + 1 to form a travel segment.\textsuperscript{19} The advantage of this method is that the actual travel segment can be better extracted from the GPS track points, the non-travel data are removed, and the characteristics of the actual travel mode are more completely and accurately presented. The limitations are mainly as follows. First, the trajectory segmentation method based on the static point can divide the travel chain more effectively, but for the inside of the travel chain, the performance of different travel segment division depends mainly on whether there is obvious stay in the transfer phase. If the traveler walks home immediately after taking the bus, there is no obvious stop in the middle, and the method cannot be effectively divided. Second, the GeoLife GPS Trajectory data set used in this article is affected by the GPS acquisition device, collecting only the GPS track points generated outdoors, and the device will only collect GPS track points if the traveler’s speed or direction changes to a certain threshold.\textsuperscript{20,21} Therefore, the parking state does not generate GPS track point records in this data set, and the trajectory segmentation based on static point method is not applicable to the data set used in this article.

**Trajectory segmentation based on GIS data of road network.** Considering that travel data are affected by external factors such as bus arrival, its main features such as speed will be confused with other travel patterns. Therefore, in addition to fully exploiting GPS track information, road network information is also considered by scholars to obtain higher accuracy rate.\textsuperscript{22} Especially for public transportation such as bus, subway, and other travel patterns, the geographical location information of bus stations and subway stations provides more powerful support for the identification of the track segment, so some scholars also integrated road network GIS data such as bus lines and subway line data.\textsuperscript{23,43}

The advantage of this method is mainly that the accuracy of track segmentation and travel pattern recognition can be improved to some extent. Especially for buses and subways, if a track has a very high degree of overlap with a bus line or a subway line, then the
travel pattern can be basically determined to be a bus or subway. However, the shortcomings of this method are the increase in the amount of data to be processed, the availability of GIS data that can be matched, the scalability of the model, etc.\textsuperscript{24} Because this method does more harm than good, although this method is popular in the early days, more scholars tend to explore the characteristics of GPS data itself to complete the travel pattern recognition problem.

**Trajectory segmentation based on transition point.** The transition point refers to different GPS track points before and after the travel segment. Scholars have adopted different techniques on this issue. Zheng et al.\textsuperscript{25} demonstrated that the switching between different travel patterns mostly involves the walking process, so they use the threshold method to pass the two characteristics of velocity and acceleration, differentiate the walking point from the non-walking point, and then merge the trajectory by setting the corresponding rules. The above method has a high dependence on the acquisition frequency of GPS track points due to the extraction and analysis of a large number of point features. Witayangkurn et al.\textsuperscript{26} proposed that high-frequency acquisition would consume a lot of battery energy, so in order to be closer to the real-life scene, based on the experiment of Zheng et al.,\textsuperscript{25} they integrated the speed conversion rate and the number of track points on the subway line as new features, the acceleration characteristics with unstable performance are removed, and the travel mode recognition is performed on the GPS data with low acquisition frequency. Different from identifying the low-speed GPS trajectory points, Yao\textsuperscript{27} believes that the travel speed will be repeatedly fluctuated due to external factors such as intersections and traffic congestion, which is similar to signal singularity characteristics; therefore, a wavelet analysis modulus maxima algorithm was introduced to identify the transition points.

The trajectory segmentation method based on transition point recognition has the following advantages. First, the transition point recognition can effectively distinguish the travel segments belonging to different travel patterns, and extract the unique travel segments in the travel pattern, which lays a solid foundation for pattern recognition. Second, compared with the unified interval method, scholar Zheng et al. have confirmed the advantages of the trajectory segmentation method based on the transformation point recognition. Many papers mention the process of the gradual decrease and increase of the switching process of most travel patterns. This feature lays the research direction for the transition point identification. Third, compared with the trajectory segmentation method based on road network GIS data, the trajectory segmentation method based on transition point recognition has weak dependence on data and strong expandability, and greatly reduces the calculation amount, which can achieve faster calculation speed.

At the same time, this method also has some drawbacks: the velocity characteristic near the transition point has a fluctuation process that first decreases and then increases. This process is similar to the characteristics of bus stop station and vehicle waiting signal,\textsuperscript{28} so there is a greater likelihood that these points will be mistaken for transition points and will need to be corrected by subsequent steps. However, all the speed-dependent trajectory segmentation methods mentioned above have this defect. Therefore, compared with other trajectory segmentation methods, the trajectory segmentation method based on transition point recognition still has its outstanding advantages.

**Travel pattern recognition**

Travel pattern recognition is based on the feature values of the extracted trajectory segments to select appropriate machine learning models for classification. In this process, the selection of eigenvalues and models is the most critical.

Speed and acceleration are the two most basic features of pattern recognition, but only the eigenvalues of maximum speed, average speed, and maximum acceleration have proven difficult to distinguish between complex and rich modes of travel. Zheng et al.\textsupersoft{4} added new features such as direction change rate, stop rate, and rate of change based on their previous research, and proved these new features have better robustness than the original basic features. Some researchers have proposed acceleration features in the vertical direction to eliminate inter-axis data interference by modulo.\textsupersoft{29,30} Considering that most travel patterns generally travel in an unobstructed space with good GPS signals, and the GPS signal in the underground space where the subway is located is weak, there may even be no signal inside the tunnel. Therefore, Li et al.\textsupersoft{31} have added features related to invalid positioning data to distinguish between subway and other travel patterns. Since the data sets used by various scholars in the experiment process are different, the feature value selection is also greatly dependent on the information provided by the data set.

In terms of model selection, the models used by scholars mainly focus on traditional machine learning models, such as SVM, decision tree, Bayesian network, and random forest. These models mainly rely on the selection of artificial features as the input to complete the classification.\textsupersoft{13,19,32} The main reasons for using this type of model are: the vast majority of public data set GPS trajectory points are large, but the number of travel segments is small, and it is unable to provide
sufficient support for deep learning. However, despite the limited amount of data, some scholars have tried to use the deep learning model to solve the problem of travel model identification. Some experimental comparisons show that if the data volume is increased, the deep learning model has the potential to achieve better recognition results. The method can avoid the disadvantages of long time spent on manual feature selection, high requirements on feature selection for industry experience, and inability to fully display travel characteristics.

The content of this article is arranged as follows. The first part introduces the research technology route and innovation point of this article. The second part summarizes the four track segmentation methods and pattern recognition problems. The third part explains the basic concept of travel pattern recognition problem. The fourth part will carry out a series of processing on the data set used in this article. The fifth part will introduce the trajectory segmentation based on transition point method and related experiments. The sixth part will introduce and use three machine learning algorithms for pattern recognition experiments. Finally, the seventh part will summarize the experimental results and make future prospects. The basic concepts of the travel pattern recognition problem are explained in the next part.

Basic concepts of travel pattern recognition

GPS trajectory points

The GPS trajectory point is information that is composed of latitude and longitude coordinates and time stamps, and can describe the spatiotemporal information of the object. The GPS trajectory point is usually represented by P_i. Figure 2 shows an example of a GPS trajectory point. On the right side in Figure 2, P_1, P_2, ..., P_n represent a series of trajectory points, and the left side is an example of the latitude and longitude of the trajectory points and timestamp data.

Trajectory, travel chain, and start and end points

A trajectory is a segment consisting of a series of consecutive GPS trajectory points {P_1, P_2, ..., P_n}. The trajectory contains all the travel records. When the time difference between two consecutive points P_i and P_{i+1} exceeds a certain range (20 min in this article), the trajectory of P_i and P_{i+1} is divided into two segments. The two trajectories are considered to be independent travel chains.

The first GPS track point of the travel chain is the start point, and the last GPS track point of the travel chain is the end point. The start point and the end point are special points in the GPS track point. Figure 3 shows an example of a trajectory, a travel chain, and start points and end points.

Travel segments and transition points

In a complete travel chain, the traveler may use one or more travel modes \{M_1, M_2, ..., M_n\} to complete the trip. When the traveler uses more than one travel mode (i.e. n > 1), M_i's last GPS track point is different from M_{i+1}'s first GPS track point travel mode, thus defining M_i's last GPS track point P_i, and n is the transition point. The transition point divides the travel into different travel segments, and the travel mode of the travel segment is unique. Figure 4 shows an example of travel segments and transition points.

Walking and non-walking sections

The walking section indicates that the travel pattern is a walking section of the walk, and the non-walking section indicates that the travel pattern is a travel section other than walking (i.e. bicycle, bus, taxi/private car). The travel pattern recognition mainly includes identifying the travel segment and identifying the specific travel pattern of the travel segment, so the basis of the travel pattern recognition is the transition point recognition. The general steps are as follows:

1. First, the GPS track points \{P_1, P_2, ..., P_n\} are sorted in chronological order to form a trajectory. When the time difference between two consecutive points P_i and P_{i+1} exceeds a certain range, a travel chain is formed. Mark the first GPS track point of the outgoing chain as the...
start point, and the last GPS track point of the outgoing chain as the end point.

2. Second, for the travel chain, the transition point is marked by the transition point recognition technique. The travel chain before and after the transition point is divided into different travel segments. Finally, the travel segment-related feature values are calculated to identify the travel pattern.

The next part will be a series of pre-processing work on the data used in this article.

**Data processing work**

**Data description**

The experimental data in this article are the public data set of the GeoLife GPS Trajectories provided by the GeoLife project from April 2007 to August 2012 by the Microsoft Research Asia. The data set was recorded by 182 volunteers. The data set consists of two parts: the GPS track point and the travel segment information with the travel pattern label. Among the GPS track point data provided by 182 volunteers, 69 recorded the corresponding travel pattern tags. This article selects the data with tags to conduct experiments.

The travel patterns covered by the data set are walking, running, bicycle, bus, taxi/private car, subway, train, airplane, ship, etc. The sample size is shown in Table 1.

Due to the small amount of data in some travel patterns (airplane, others), it is impossible to achieve a better recognition effect and is not included in the scope of this study. In addition, because the GPS acquisition equipment is in the state of signal loss in the underground, the way of travel marked as subway is actually light rail, not for the subway. Individual differences in

### Table 1. Total distance and total duration of each travel pattern.

| Travel pattern     | Total distance (km) | Total duration (h) |
|--------------------|---------------------|--------------------|
| Walking            | 10,123              | 5460               |
| Bicycle            | 6495                | 2410               |
| Bus                | 20,281              | 1507               |
| Taxi/private car   | 32,866              | 2384               |
| Train              | 36,253              | 745                |
| Airplane           | 24,789              | 40                 |
| Others             | 9493                | 404                |
volunteers may cause the light rail to be marked as either train or subway. Considering various factors, the travel patterns within the scope of the study were determined to be walking, bicycles, buses, and taxis/private cars. The data set field characteristics are shown in Tables 2 and 3.

### Data pre-processing

**Data set field filtering and generation.** Since the altitude has no effect on the experimental operation, the altitude field in Table 2 is removed. The role of the date field in Table 4 is equivalent to the role of date as a string and time as a string field. To facilitate experimental operation and data understanding, the fields of date as a string and time as a string are named as time stamp field instead of date field.

After the field is selected, the GPS track point data set data fields are as shown in Table 5.

**Data deduplication.** Considering that the travel segment label is calibrated by volunteers, many studies using this data set have raised the problem of certain repetitive calibrations, so the travel segment data is deduplicated to obtain the only valid data. In addition, due to the failure of the GPS acquisition device, some GPS data collected will have data duplication problems. As shown in Table 6, the GPS track point data with the same latitude, longitude, and time stamp are deduplicated.

**Label correspondence.** The data of Tables 5 and 6 are correspondingly assigned, and the segment tags are assigned to the GPS track points whose time is within the range of the travel segment, thereby obtaining the travel tags corresponding to each GPS data point. At the same time, according to the sequence, the first GPS track point and the last GPS track point in the range of the travel section are, respectively, marked with a start point and an end point. The corresponding data format is shown in Table 7.

**Data screening.** Since the signal quality of the GPS device is affected by the outside world, some GPS data collected will be sparse. Because this part of the GPS data quality problem will cause the two GPS track points to be separated and the distance is far away, only the straight-line distance can be obtained during the

### Table 2. GPS track point data set field characteristics.

| Field name | Meaning | Value |
|------------|---------|-------|
| Latitude   | The latitude of the GPS track point | Number value, such as 39.906631 |
| Longitude  | The longitude of the GPS track point | Number value, such as 116.385564 |
| Altitude   | The altitude at which the GPS track point is located | Number value, such as 492. – 777 means invalid |
| Date       | Number of days from 1899/12/30 | Number value, such as 40097.5864583333 |
| Date as a string | Date | String, such as 2009-10-11 |
| Time as a string | Time | String, such as 14:04:30 |

GPS: Global Positioning System.

### Table 3. Travel pattern label data set.

| Field name | Meaning | Value |
|------------|---------|-------|
| Start time | The start time of travel segment | Character, such as 2008/04/02 11:24:21 |
| End time   | The end time of travel segment | Character, such as 2008/04/02 11:50:45 |
| Transportation mode | The travel pattern | Character, such as walk |

### Table 4. Time stamp field generation.

| Date as a string | Time as a string | Time stamp |
|------------------|------------------|------------|
| 2008-10-23       | 02:53:04         | 2008-10-23 02:53:04 |
| 2008-10-23       | 02:53:05         | 2008-10-23 02:53:05 |
| ...              | ...              | ...        |

### Table 5. GPS track point data set data field after field selection.

| Field name | Meaning | Value |
|------------|---------|-------|
| Latitude   | The latitude of the GPS track point | Number value, such as 39.906631 |
| Longitude  | The longitude of the GPS track point | Number value, such as 116.385564 |
| Time stamp | Time stamp | Character, such as 2008-10-23 14:04:30 |

GPS: Global Positioning System.
calculation. Therefore, this article counts the average GPS sampling rate of each travel segment, and removes the travel segment of a GPS track point after the average GPS sampling rate exceeds 10 s. In addition, because part of the travel segment records is incomplete, the duration is short, and the information covered is too small. In this article, the travel segment with a travel duration of less than 30 s and a GPS track record number \( N_{\log} \) of less than or equal to 5 is deleted and removed. After screening, the data volume information of each travel pattern is shown in Table 8. The private car and taxi data are small and the travel characteristics are similar, and they are merged into one category.

**Basic eigenvalue calculation**

In order to extract richer travel characteristics through the latitude, longitude, and time stamp information of GPS track points, the feature values are selected and defined and calculated. The basic eigenvalues include distance between two points, time difference, instantaneous velocity, and instantaneous acceleration.\(^4\) The definition and calculation formula are as follows:

1. **Distance between two points** \( d_{i,i+1} \)

Based on the latitude and longitude information of two adjacent GPS track points, the approximate distance between the two points can be calculated using the Haversine formula, in kilometers:

\[
 d_{i,i+1} = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_{i+1} - \phi_i}{2} \right) + \cos(\phi_i) \cos(\phi_{i+1}) \sin^2 \left( \frac{\lambda_{i+1} - \lambda_i}{2} \right)} \right)
\]

(1)

where \( d_{i,i+1} \) represents the distance between two adjacent GPS track points, \( r \) is the radius of the earth, taking 4885.936406 km, \( \phi_i \) and \( \phi_{i+1} \) represent the latitude of the GPS trajectory points \( i \) and \( i + 1 \), respectively, and \( \lambda_i \) and \( \lambda_{i+1} \) represent the longitude of the GPS trajectory points \( i \) and \( i + 1 \), respectively.

2. **Time difference** \( \Delta t_{i,i+1} \)

According to the time stamp information of two adjacent GPS track points, the time difference \( \Delta t \) can be calculated and recorded in seconds. The definition formula is as follows:

\[
 \Delta t_{i,i+1} = T_{i+1} - T_i
\]

(2)

where \( T_i \) is the time stamp of the GPS track point \( i \).

3. **Instantaneous speed** \( v_{i,i+1} \)

Instantaneous velocity describes the velocity of an object at a certain time or through a certain position,

---

**Table 6. GPS track point data repetition.**

| Latitude       | Longitude        | Time stamp          |
|----------------|------------------|---------------------|
| 40.005148      | 116.321867       | 2008-06-27 01:17:40 |
| 40.005148      | 116.321867       | 2008-06-27 01:17:40 |
| 40.005117999999996 | 116.32186499999999 | 2008-06-27 01:17:41 |
| 40.005117999999996 | 116.32186499999999 | 2008-06-27 01:17:41 |

GPS: Global Positioning System.

**Table 7. Data format after label correspondence.**

| Actual mode | Latitude | Longitude | Time stamp      | Point feature |
|-------------|----------|-----------|-----------------|---------------|
| Walk        | 39.477125| 75.989985 | 2008-04-02 06:09:26 | Start point   |
| Walk        | 39.477083| 75.989999 | 2008-04-02 06:09:28 |               |
| Walk        | 39.473308| 75.987955 | 2008-04-02 06:28:25 | Change point  |
| Car         | 39.473332| 75.988222 | 2008-04-02 06:30:57 |               |
| Car         | 39.474775| 75.980305 | 2008-04-02 06:34:20 | End point     |

**Table 8. Data volume of each travel mode after screening.**

| Travel pattern | Data volume | Total data volume |
|----------------|-------------|-------------------|
| Walk           | 3489        | 3489              |
| Bicycle        | 1285        | 1285              |
| Bus            | 1719        | 1719              |
| Private car    | 665         | 1077              |
| Taxi           | 412         | 412               |
| Total          | 7570        | 7570              |
representing the displacement of the object in an infinitely short time. Since the acquisition time difference between two adjacent GPS track points under normal conditions is very short and mostly concentrated within 5 s, the speed value calculated by formula (3) can be approximated as the instantaneous speed in m/s. It is calculated as follows

\[ v_{i+1} = \frac{d_{i+1}}{\Delta t_{i+1}} \quad (3) \]

where \( v_{i+1} \) is the instantaneous speed from the GPS track point \( i \) to the GPS track point \( i + 1 \).

4. Instantaneous acceleration \( a_{i+1} \)

Acceleration is the physical quantity that reflects the speed of the object's speed change. The instantaneous acceleration describes the degree of change in speed per unit time. When the distance between the two points is short enough, the motion process can be approximated as a uniform acceleration linear motion. The acceleration can be calculated by formula (4), the unit is m/s², and the definition formula is as follows

\[ a_{i+1} = \frac{v_{i+1} - v_{i-1}}{\Delta t_{i}} \quad (4) \]

where \( a_{i+1} \) is the instantaneous acceleration from the GPS track point \( i \) to the GPS track point \( i + 1 \).

5. Direction angle \( H_{i+1} \)

The direction angle is the angle between the direction of the track from the GPS track point \( i \) to the GPS track point \( i + 1 \) and the north direction, which can be calculated by equation (5):

\[ H_{i+1} = \tan^{-1} \left( \frac{\sin(\lambda_{i+1} - \lambda_{i})}{\cos(\varphi_{i}) \cdot \tan(\varphi_{i+1}) - \sin(\varphi_{i}) \cdot \cos(\lambda_{i+1} - \lambda_{i})} \right) \quad (5) \]

where \( \varphi_{i} \) and \( \varphi_{i+1} \) represent the latitude of the GPS trajectory points \( i \) and \( i + 1 \), respectively, and \( \lambda_{i} \) and \( \lambda_{i+1} \) represent the longitude of the GPS trajectory points \( i \) and \( i + 1 \), respectively.

After the data are processed, the next part will describe the track segmentation method based on the conversion point technique used in this article.

**Track segmentation method based on transition point recognition**

This section will mainly introduce the trajectory segmentation method based on the transition point recognition technology. First, compare the performance of different experimental schemes on the identification of walking and non-walking points. Then, on the basis of the optimal scheme, the walking and non-walking sections are further extracted and corrected, and different extraction and correction rules are compared. Finally, the GPS track points alternated between the two outgoing segments are identified as the transition points, and compared with the real transition points. The track segmentation results are evaluated and discussed by recall rate and precision rate.

**Walking and non-walking point identification**

This article first calculates the transition probability between different travel patterns. The calculation formula is shown in formula (6)

\[ P_{mj} = \frac{N(mj)}{N(m)} \quad (6) \]

where \( P_{mj} \) is the probability of transition from the travel pattern \( i \) to the travel pattern \( j \), \( N(mj) \) is the number of times the travel pattern \( i \) is converted to the travel pattern \( j \), and \( N(m) \) is the number of times the total travel pattern transition occurs.

The results are shown in Table 9. More than 85% of the travel pattern transitions experience a walking process, so the first and last GPS track points of the walking segment have a greater likelihood of becoming transition points. In this article, we first distinguish between walking and non-walking points, and then extract the walking and non-walking segments, determine the position of the transition point, and evaluate the results.

Walking and non-walking point identification is the basic work of conversion point identification, and its accuracy has a great influence on subsequent experiments. This article selects three kinds of walking and non-walking point recognition methods: based on threshold setting, based on multi-layer perceptron and based on DBSCAN clustering, and compares their performances.

**Walking and non-walking point recognition based on threshold setting.** In this article, the absolute values of the speed and acceleration of the walking and non-walking GPS track points are counted. The percentiles are shown in Tables 10 and 11.

It can be seen from the above table that the difference in the absolute value distribution of the

### Table 9. Travel pattern transition probability statistical matrix.

|       | Walk | Bike | Bus | Car (Taxi) |
|-------|------|------|-----|------------|
| Walk  | 60%  | 3%   | 1%  | 0.1%       |
| Bike  | 88%  | 0%   | 1%  | 0.2%       |
| Bus   | 39%  | 0.2% | 2%  | 1%         |
| Car (Taxi) | 12%  | 0.2% | 0.2%| 0.1%       |
| Total | 57.8%| 4.2% | 28.7%| 9.3%       |
acceleration between the walking and non-walking points is not significant, but the speed distribution of the two is quite different. Based on the 70% quantile of the walking point of 1.5383 m/s and the non-walking point of 25% of the 1.5777 m/s, it can be seen that the instantaneous speed of 70% of the walking points is less than 1.5383 m/s, and only 25% of the non-walking point instantaneous speed is less than 1.5777 m/s. Therefore, this article sets the speed threshold to distinguish between walking and non-walking points. The bold values in Table 10 and Table 11 show that the value distribution of 50% - 95% quantile of walking point is close to that of 20% - 50% quantile of non-walking point. Therefore, when setting parameters, the speed threshold interval is set as [1,4], and the step size is 0.5.

### Table 10. Walking point speed, acceleration percentile.

| Percentile | 10% | 30% | 50% | 70% | 80% | 85% | 90% | 95% | 100% |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| | (m/s) | | | | | | | | |
| | 0.1424 | 0.7330 | 1.1344 | 1.5383 | 1.8615 | 2.137 | 2.6408 | 3.9849 | 531.2386 |
| | (m/s²) | | | | | | | | |
| | 0.0102 | 0.0603 | 0.1437 | 0.2897 | 0.4264 | 0.5383 | 0.7201 | 1.1347 | 499.8220 |

### Table 11. Non-walking point speed, acceleration percentile.

| Percentile | 10% | 15% | 20% | 25% | 30% | 50% | 70% | 90% | 100% |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| | (m/s) | | | | | | | | |
| | 0.3548 | 0.7452 | 1.1549 | 1.5777 | 2.0782 | 4.0009 | 7.7748 | 14.7259 | 1253.2140 |
| | (m/s²) | | | | | | | | |
| | 0.0239 | 0.0412 | 0.0616 | 0.0851 | 0.1103 | 0.2390 | 0.4499 | 0.9860 | 626.2410 |

Walk and non-walking point recognition based on multi-layer perceptron. The principle of multi-layer perceptron is given in Yang. The input of the constructed multi-layer perceptron is the GPS track point with label (walking, non-walking) and the velocity and acceleration characteristics. At the same time, the parameters of the hidden layer, the number of neurons, and the activation function are adjusted. The classification accuracy rate and the loss value are used as evaluation indexes, and finally, the label classification result of the GPS track point is output.

Walking and non-walking point recognition based on DBSCAN clustering. DBSCAN clustering uses homogeneity and completeness as evaluation indicators to achieve the best effect of making the sample distance within the cluster as close as possible and the distance between the clusters as far as possible. The two indicators are calculated as

\[
\text{Homogeneity} = 1 - \frac{\sum_{k=1}^{|K|} \sum_{c=1}^{|C|} \frac{n_{k,c}}{n} \cdot \log \left( \frac{n_{k,c}}{n} \right)}{\sum_{c=1}^{|C|} \frac{n_c}{n} \cdot \log \left( \frac{n_c}{n} \right)}
\]

\[
\text{Completeness} = 1 - \frac{\sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \cdot \log \left( \frac{n_{c,k}}{n} \right)}{\sum_{k=1}^{|K|} \frac{n_k}{n} \cdot \log \left( \frac{n_k}{n} \right)}
\]

where \( n \) is the total number of samples, \( n_c, n_k \) represents the number of samples belonging to class \( c \) and cluster \( k \), respectively, and \( n_{c,k} \) represents the number of samples belonging to class \( c \) and clustered into cluster \( k \). The homogeneity represents whether a cluster formed by DBSCAN clustering only contains the same data points, and the completeness represents whether the data points with the actual label consistency are divided into clusters by DBSCAN clustering.

The DBSCAN clustering in this article only uses homogeneity as the evaluation index. The main reasons are as follows: most of the GeoLife GPS Trajectory data sets selected in this article collect GPS tracking points with a fixed frequency of 1–5 s. Therefore, the walking point presents a denser distribution than the non-walking point. Figure 5 is a trajectory diagram of a travel chain that is converted into a driving. It can be seen that the GPS trajectory distribution of the walking section is much denser than that of the non-walking section. The distribution characteristics of GPS track points near the transition point have undergone significant changes. Therefore, this article borrows the density-based characteristics of DBSCAN clustering, and distinguishes between walking and non-walking points, which is essentially a two-class problem.

The output of the binary classification problem should be the points (i.e. walking points) and the noise points (i.e. non-walking points) that are clustered in the cluster. Therefore, no matter how many clusters are formed in the DBSCAN cluster, the actual labels of the points in the cluster are actually consistent. The completeness represents whether the data points with the actual label consistency are divided into a cluster by DBSCAN clustering. In this problem, the walking points do not need to be clustered in one cluster, so the evaluation index completeness is not applicable in this
study. The homogeneity represents whether a cluster formed by DBSCAN clustering only contains the same data points. In this study, the true label of a cluster should be walking, so the homogeneity of the index is in this study is still applicable.

Because DBSCAN clustering based on density-based characteristics to solve the two-category problem, the final result of evaluating its performance is still the accuracy of its classification. Therefore, the classification accuracy rate is used as the main evaluation index, and the homogeneity is used to prove the performance of DBSCAN clustering. The experimental scheme is as follows: the model input is GPS track point data with a label (walk, non-walk), and the adjustable parameter is a distance threshold $e$ of a certain point field and a threshold $MinPts$ of the number of samples in a certain point field, wherein the range value of $e$ is $[5, 10, 15, 20, 25, 30]$ and the range of $MinPts$ is $[5, 10, 15, 20, 25, 30]$. The classification accuracy and homogeneity were used as evaluation indicators. The model output is the DBSCAN clustering result under optimal parameters ($-1$ is noise, the other is cluster, such as the $n$th cluster label is $n$). The noise point label is named as non-walk, and the other cluster point label is named as walk.

**Comparison and analysis of experimental results.** When the walking and non-walking point identification scheme based on the threshold setting is selected, the experimental results are as follows: when the speed threshold $C_v$ is gradually increased from 1 to 2, the classification accuracy rate gradually increases, and thereafter shows a downward trend. When $C_v$ equals 2, the classification accuracy is the highest.

When using the multi-layer perceptron–based walking and non-walking point identification scheme, the specific experimental scheme is as follows: the multi-layer perceptron is built using the sequential model of Keras. Use the sigmoid as the activation function. Adjustable parameters include the number of hidden layers, the number of neurons in each layer, and the activation function. In order to determine the optimal number of iterations, this article selects the EarlyStopping function in Keras. Taking the validation set loss as the reference, if the validation set loss is not improved within two iterations, the training is stopped, and the improvement amount $0.0001$ is no improvement. The structure of each model and its performance are shown in Table 12.

Through comprehensive comparison, it can be found that in the models 1, 2, and 3, as the number of hidden layers increases, the performance of the model shows a decreasing trend. Considering that the input of the problem itself is relatively simple, increasing the network depth will affect the performance of the model. Therefore, a multi-layer perceptron with a hidden layer number of 1 is more suitable for this problem. By comparing the models 1, 4, and 5, it can be seen that the change in the number of neurons generally has little effect on the performance of the model. Considering comprehensively, in order to reduce the complexity of the model and speed up the training of the model, the model with the best performance is model 4 (as shown with bold values in Table 12).

When using the walking and non-walking point identification scheme based on DBSCAN clustering, the accuracy of each parameter and its corresponding classification result are shown in Table 13.

It can be seen from Table 13 that when $e$ is gradually increased, $MinPts$ can only continue to maintain a superior recognition result as it becomes larger. If $e$ increases and $MinPts$ does not change, then more points are identified as noise (i.e. non-walking points).
It can be seen from the bold values in Table 13 that the optimal combination of the two relations basically appears at the diagonal of Table 14, that is, $E = \text{MinPts}$, and as $e$ increases, the correct rate of DBSCAN clustering also increases. Therefore, this article further sets up a new parameter pool ($E = \text{MinPts} \times 2[50,300]$, step = 50) to conduct experiments. When $e = \text{MinPts} = 200$, the DBSCAN clustering effect is optimal, and the classification accuracy rate reaches 0.822. Taking the path chain of Figure 5 as an example under this parameter, the visualization result after DBSCAN clustering is shown in Figure 6. The color point in the figure is the cluster formed by the DBSCAN algorithm (i.e. the walking point), and the gray point is the noise point (i.e. non-walking point).

It can be seen from Figure 6 that the DBSCAN algorithm performs better in the walking and non-walking point classification of this travel chain, and only a small number of clusters (i.e. walking points) appear in the non-walking section, which may be due to other external environments such as traffic lights intersections, road congestion, and other reasons. However, only a few points in the walking segment are identified as noise points (i.e. non-walking points), and a plurality of clusters can be found in a walking section. The reason may be that the signal of the GPS mobile phone device is temporarily missing, so that the distance between the two points is large. It is also possible that there are partially adjacent walking GPS track points due to the large instantaneous velocity, resulting in a distance between the two points exceeding the threshold of the current parameter.

The final experimental results of the three experimental schemes are as follows: the optimal parameter based on the threshold scheme is $C_v = 2$, and the classification accuracy is 74.64%. The optimal parameters

| Table 12. Classification accuracy of each model structure of multi-layer perceptron. |
|---------------------------------|--------------|--------------|--------------|--------------|--------------|
| Model 1 | Model 2 | Model 3 | Model 4 | Model 5 |
| Number of hidden layers | 1 | 3 | 5 | 1 | 1 |
| Number of neurons | 8 | 8 | 8 | 4 | 16 |
| Best iterations | 12 | 14 | 4 | 7 | 12 |
| Model performance | Convergence smooth | Convergence concussion | Convergence concussion | Convergence smooth | Convergence smooth |
| Accuracy of validation set | 0.7599 | 0.7627 | 0.6761 | 0.7595 | 0.7575 |

| Table 13. $e \in [5, 30]$, MinPts $\in [5, 30]$, step = 5, DBSCAN clustering accuracy. |
|-----------------|--------------|--------------|--------------|--------------|--------------|
| MinPts/e | 5 | 10 | 15 | 20 | 25 | 30 |
| 5 | 0.7209 | 0.686 | 0.6118 | 0.5603 | 0.5218 | 0.49 |
| 10 | 0.6934 | 0.7273 | 0.7274 | 0.6934 | 0.6544 | 0.6168 |
| 15 | 0.6903 | 0.7217 | 0.7399 | 0.7451 | 0.7254 | 0.6981 |
| 20 | 0.6925 | 0.7143 | 0.7381 | 0.7508 | 0.7564 | 0.7469 |
| 25 | 0.6946 | 0.7112 | 0.7343 | 0.7508 | 0.7603 | 0.7659 |
| 30 | 0.6965 | 0.7118 | 0.7291 | 0.7488 | 0.7603 | 0.7688 |

| Table 14. Example of walking and non-walking point identification output data based on DBSCAN clustering. |
|---------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Actual mode | Latitude | Longitude | Time stamp | Point feature | Predict mode | Predict result | delta d | delta t |
| Walk | 39.47713 | 75.98999 | 2008-04-02 06:09:26 | Start point | Non-walk | FALSE | 4.6828 | 2 |
| Walk | 39.47708 | 75.98999 | 2008-04-02 06:09:28 | Walk | TRUE | 3.4401 | 1 |
| Walk | 39.47705 | 75.98998 | 2008-04-02 06:09:29 | Walk | TRUE | 2.3247 | 1 |
| Walk | 39.47703 | 75.98997 | 2008-04-02 06:09:30 | Walk | TRUE | 6.2465 | 38 |
| Walk | 39.47698 | 75.98997 | 2008-04-02 06:10:08 | Walk | TRUE | 5.9498 | 2 |
| Walk | 39.47693 | 75.98994 | 2008-04-02 06:10:10 | Walk | TRUE | 1.6654 | 1 |
| Walk | 39.47691 | 75.98994 | 2008-04-02 06:10:11 | Walk | TRUE | 0.9525 | 1 |
| Walk | 39.47691 | 75.98993 | 2008-04-02 06:10:12 | Walk | TRUE | 2.1607 | 1 |

DBSCAN: density-based spatial clustering of applications with noise.
based on the multi-layer perceptron scheme are hidden layer = 1, neuron = 4, and the classification accuracy rate is 75.95%. The optimal parameters based on DBSCAN clustering scheme are $\varepsilon = 200$, MinPts = 200, the classification accuracy is 82.20%, and the homogeneity is 0.85. It can be seen from the results that the DBSCAN clustering effect is better under the optimal parameters of the three experimental schemes. At the same time, the higher homogeneity score also supports this conclusion.

Compared with DBSCAN clustering, the threshold accuracy and multi-layer perceptron classification accuracy are lower, probably because it depends only on the instantaneous velocity or instantaneous acceleration characteristics, and the two calculations are affected by the radius of the earth in the Haversine formula. The return value itself is in km, so it is easy to cause errors of 10 m or even 100 m. However, the GPS track point has drift and deviation caused by the low precision of the acquisition equipment, and the intermediate calculation further expands the error, which affects the classification accuracy. In addition, these two methods classify each GPS track point as an independent individual and are therefore more susceptible to external factors. The DBSCAN clustering takes into account the number of trajectory points that satisfy the condition, and considers the continuous state between geographical distributions, so it has better anti-noise effect on external factors.

Therefore, in this article, the walking and non-walking point identification scheme based on DBSCAN clustering is selected, and the distance interval $\Delta d$ and time difference $\Delta t$ of each GPS track point from its next GPS track point are calculated. The final output data are shown in Table 14, where the predict mode field is the travel pattern identified by the DBSCAN cluster, and the predict result is whether the recognition result is correct.

After completing the walking and non-walking recognition of the track points, the walking and non-walking sections will be extracted below.

**Walking and non-walking section extraction**

Based on the above-mentioned walking point and non-walking point recognition results, this article extracts the walking and non-walking sections by means of merging and correction, identifies the transition point, and evaluates the recognition result by recall rate and accuracy. If the time difference between the identified transition point and the actual transition point is within 2 min, the recognition result is considered correct.

**GPS track point combination.** If the GPS track points $n - i, n, n + i$ are the same in the predict mode field in the walking point and non-walking point identification, the first GPS track point $n - i$ is used as the start point, and the last GPS track point $n + i$ is used as the end point. The time stamp and latitude and longitude information of the start point termination point are also separately recorded, and the section is assigned a walk or non-walk. In the merging process, if the predict mode values of the GPS track points $i$ and $i + 1$ are the same, but the $\Delta t$ field of the point $i$ takes a value greater than 20 min, it is still divided into two segments, with $i$ as the end point of the first travel segment and $i + 1$ as the start point of the second travel segment. The merge process is shown in Figure 7.

During the merging process, the number of GPS trajectory points (N_log), total duration (duration), and total distance (distance) in each interval are counted. The combined data are shown in Table 15, where the start latitude, start longitude, end latitude, and end longitude fields are replaced by ellipsis because of limited display space.
Travel segment consolidation and correction. After the first merger, the duration and distance of the travel section are short, which is quite different from the actual travel distance. Taking Table 15 as an example, the first non-walking segment contains only one GPS track point, and both duration and distance are 0. This is due to the misjudgment of some points during the recognition of the walking point and the non-walking point. Therefore, this article further sets rules to merge and correct these shorter segments.

By selecting small sample data, this article designs and compares the following three experimental schemes to merge the sections whose distance is less than or equal to the threshold \( C_{\text{distance}} \), and the \( \delta t \) from the next trip section is less than 20 min:

1. Select the section label with the longest distance in the interval as the entire segment label. The accuracy rate is 77.35%.
2. Compare the total length of the walk section and the non-walk section distance in the interval, and take the larger one as the entire segment label. The accuracy rate is 78.73%.
3. Compare the number of walking and non-walk GPS track points in the interval (i.e. \( N_{\text{log}} \)), and take the larger one as the entire segment label. The accuracy rate is 80.21%.

Comparing the three schemes and the experimental results, it can be found that the experimental results of the scheme 3 are optimal. In the results of Experiments 1 and 2, most of the misidentified travel segments were actually labeled as walk, but were predicted to be non-walk after the merge. The result may be that the travel speed of the non-walk segment is faster, so the distance traveled over the same time is longer. If the actual label of a travel section is walk, which covers a small section consisting of some faster speed points, in the first and second schemes with the driving distance (distance) characteristic as the evaluation standard, the label of this section is very likely to be determined as the entire segment label, resulting in misidentification. The third scheme considers the number of walk and non-walk GPS track points in the whole interval, which avoids the disadvantages of non-walking GPS track points.

Table 15. Example of output data after GPS track point combination.

| Start time | ... | End time | ... | Predict mode | \( N_{\text{log}} \) | Duration | Distance | \( \delta t \) | \( \delta d \) |
|------------|-----|----------|-----|--------------|--------|----------|----------|--------|--------|
| 2008-04-02 06:09:26 | 2008-04-02 06:09:26 | Non-walk | 1 | 0 | 0 | 2 | 4.68 |
| 2008-04-02 06:09:28 | 2008-04-02 06:32:05 | Walk | 628 | 1357 | 928 | 1 | 4.57 |
| 2008-04-02 06:32:06 | 2008-04-02 06:34:20 | Non-walk | 109 | 134 | 916 | 17,401 | 640.43 |
| 2008-04-02 11:24:21 | 2008-04-02 11:50:45 | Non-walk | 266 | 1584 | 7332 | 93,879 | 635,260.29 |
| 2008-04-03 13:55:24 | 2008-04-03 14:01:49 | Non-walk | 384 | 385 | 4909 | 51,936 | 508,443.45 |
| 2008-04-04 04:27:25 | 2008-04-04 04:31:34 | Walk | 115 | 249 | 209 | 1 | 4.56 |
| 2008-04-04 04:31:35 | 2008-04-04 05:03:35 | Non-walk | 1918 | 1920 | 52,061 | 7995 | 57.69 |

GPS: Global Positioning System.
and long driving distance, so the experimental results are better.

Therefore, based on the third scheme, the optimal threshold $C_{\text{distance}}$ is selected and the parameter pool is set to $[50, 100, 150, 200]$. If the transition point identified by the experiment is within 2 min of the neighborhood near the true transition point, the recognition is considered correct. According to this standard, the recall rate and accuracy of transition points are compared under each threshold. The results are shown in Table 16, and the trend is shown in Figure 8.

**Table 16.** $C_{\text{distance}} \in [50, 100, 150, 200]$, recall rate, and accuracy result of transition points.

| $C_{\text{distance}}$ | 50  | 100 | 150 | 200 |
|-----------------------|-----|-----|-----|-----|
| Recall rate           | 0.86| 0.86| 0.84| 0.83|
| Precision rate        | 0.53| 0.58| 0.66| 0.59|

**Figure 8.** $C_{\text{distance}} \in [50, 100, 150, 200]$, the trend of transition point recall rate, and accuracy.

and long driving distance, so the experimental results are better.

Therefore, based on the third scheme, the optimal threshold $C_{\text{distance}}$ is selected and the parameter pool is set to $[50, 100, 150, 200]$. If the transition point identified by the experiment is within 2 min of the neighborhood near the true transition point, the recognition is considered correct. According to this standard, the recall rate and accuracy of transition points are compared under each threshold. The results are shown in Table 16, and the trend is shown in Figure 8.

**Analysis of conversion point recognition results**

According to the experimental results in Figure 8, when the segmentation threshold $C_{\text{distance}}$ is gradually increased, the conversion point recall rate shows a slight downward trend, and the accuracy rate is gradually increased. The reason is mainly because when the $C_{\text{distance}}$ threshold is increased, part of the travel segment with a shorter actual travel distance will be merged into other travel segments during the merge process. When the two travel segments are merged as one segment, the transition point also disappears with the merge, so it cannot be recognized, which leads to a decrease in the recall rate. However, as $C_{\text{distance}}$ increases, the recognition rate of the transition point shows a 2% drop when it is increased to 150 m, and its accuracy is 13% higher than when $C_{\text{distance}} = 50$. This further illustrates that there is only a relatively low proportion of real short-distance travels in the shorter distance segments before the merger and correction, and most of the rest are due to the inaccurate identification of walking and non-walking points. Therefore, it is necessary to correct it by setting the merge rule and the effect is remarkable.

Therefore, this article finally selects $C_{\text{distance}} = 150$ as the optimal threshold to complete the trajectory segmentation. The trajectory segmentation form after extracting the transition point is shown in Table 17, where the $N_{\log}$ field represents the number of GPS track points in this travel segment.

After the trajectory segmentation is completed, three models will be used for travel pattern recognition.

**Travel pattern recognition**

This section will first explore the intrinsic characteristics of different travel modes to select more distinctive feature values, and improve the accuracy of late travel pattern recognition. Thereafter, this section will compare the differences in the performance of different machine learning models on the GeoLife GPS Trajectory data set to identify four travel patterns: walk, bicycle, bus, and private car.

**Analysis of different travel pattern characteristics**

In this article, four sections of the travel pattern of walking, bicycle, bus, and private car are selected, and the speed curve along with the driving distance is

**Table 17.** Data format after completion of track segmentation.

| Start time ... | End time ... | Predict mode | $N_{\log}$ | Duration | Distance |
|----------------|--------------|--------------|------------|----------|----------|
| 2008-04-02 06:09:26 | 2008-04-02 06:32:05 | Walk | 629 | 1359 | 933.1175 |
| 2008-04-02 06:32:06 | 2008-04-02 06:34:20 | Non-walk | 109 | 134 | 915.9117 |
| 2008-04-02 11:24:21 | 2008-04-02 11:50:45 | Non-walk | 266 | 158 | 7332.334 |
| 2008-04-03 13:55:24 | 2008-04-03 14:01:49 | Non-walk | 384 | 385 | 4908.745 |
| 2008-04-04 04:27:25 | 2008-04-04 04:31:34 | Walk | 115 | 249 | 209.1051 |
| 2008-04-04 04:31:35 | 2008-04-04 05:03:35 | Non-walk | 1918 | 1920 | 52,061.3 |
| 2008-04-04 06:16:50 | 2008-04-04 07:51:47 | Non-walk | 1068 | 2097 | 51,272.07 |
| 2008-04-04 07:51:48 | 2008-04-04 07:53:00 | Walk | 72 | 72 | 424.1935 |
| 2008-05-18 07:24:54 | 2008-05-18 08:03:44 | Non-walk | 2134 | 2330 | 29,998.62 |
visualized. The result is shown in Figure 9. The abscissa is the driving distance and the ordinate is instantaneous speed, where the red line represents the average speed within the line segment.

It can be seen from Figure 9 that the travel distance and speed distribution of the different travel patterns are quite different. Because the overall speed of walking is slow, the travel distance of the walking section is usually limited by the physical strength of the traveler. The distance of the bicycle is longer than that of the walking. In contrast, buses and private cars have a relatively high speed and can travel a long distance in the same time. Therefore, the distance of the travel segment (trajectory length) is selected as one of the characteristics of different travel patterns.

Second, from the perspective of average speed, there is a significant difference between motor vehicles and non-motor vehicles. Usually, walking < bike < bus < taxi/private car, this conclusion is closer to the actual life. Therefore, this article selects the average instantaneous speed \( \bar{v} \) as one of the characteristics of different travel patterns.

From the distribution of instantaneous speed, there are also significant differences between motor vehicles and non-motor vehicles. The stopping frequency of the non-motor vehicle with the speed close to 0 during driving is obviously higher than that of the motor vehicle. Therefore, the stopping rate (SR) is selected as one of the characteristics of different travel patterns. The calculation formula of the SR is shown in formula (9):

\[
SR = \frac{N(v < c_v)}{\text{Trajectory length}} \tag{9}
\]

where \( N(v < c_v) \) is the number of GPS track points whose speed is less than the threshold \( c_v \), and the \( \text{Trajectory length} \) is the distance of the travel segment.

However, the speed is greatly affected by the outside world and the robustness is poor. Therefore, if the different travel modes are distinguished only from the eigenvalue of the average speed, the probability of false positive is higher. As shown in Figure 10, the average speed of the two-segment private car travel section is only about 2 m/s, similar to the performance of the pedestrian travel section on this feature, may be due to long-term slow travel caused by road congestion. However, even though the average speed \( \bar{v} \) characteristics are similar, unlike the ordinary walking section, the instantaneous speed of the private car travel section is still relatively large, exceeding 10 m/s. From the perspective of real life, the instantaneous speed of pedestrians generally cannot match this value. Therefore, in order to more fully describe the speed information of the travel segment, this article also considers the maximum instantaneous speed and selects the three

![Figure 9. Speed versus travel distance curve in different travel patterns.](image-url)
maximum instantaneous speeds as the eigenvalues to reduce the impact of the outliers. In addition, the speed change caused by the external environment is quite different from the speed under normal conditions of the travel pattern, and the motor vehicle is susceptible to such changes due to external factors, so the overall variance of the instantaneous speed is larger than that of the non-motor vehicle. Therefore, this article also selects the instantaneous velocity variance $D_v$ to further comprehensively characterize the velocity characteristics.

In addition to analyzing the speed characteristics of different travel patterns, this article also analyzes its acceleration characteristics. The results are shown in Figure 11. The abscissa is the travel distance and the ordinate is the instantaneous acceleration, where the red line represents the average acceleration within the travel segment.

As can be seen from Figure 11, since most of the travel segment acceleration and deceleration processes coexist, it can be found that the average accelerations of the different travel patterns is close to zero. In this article, the instantaneous acceleration is corrected to the absolute value of the instantaneous acceleration, and the analysis is performed again. The corrected result is shown in Figure 12.

It can be seen from Figure 12 that the instantaneous acceleration of each travel pattern is not obvious from the mean value or the maximum value. Since the four segments are randomly selected and do not have universal representation, in order to confirm the above conjecture, this article calculates the absolute value distribution of instantaneous acceleration of all segments. The result is shown in Figure 13.

By observing the percentage distribution of the absolute value of the acceleration in Figure 13, it is further confirmed that different travel patterns behave similarly in terms of acceleration and its absolute value characteristics, and may not be effectively distinguished. Therefore, the maximum value of the absolute value of the acceleration is taken as the undetermined eigenvalue and the experimental comparison is performed.

**Feature value selection**

The definition and calculation of the basic eigenvalues have been proposed in section “Basic eigenvalue calculation.” Through the analysis of the travel characteristics of different travel patterns in section “Data description,” the eigenvalues listed in Table 18 are also selected to distinguish different travel patterns. Among them, heading change rate (HCR) and velocity change rate (VCR) are obtained by Zheng et al.,\textsuperscript{12} and the calculation is as shown in formulas (10) and (11).

$$\text{HCR} = \frac{N(HC>c_{HC})}{\text{Trajectory length}}, \quad HC = |H_1 - H_2| \quad (10)$$

where $HC$ represents a change in direction, $N(HC>c_{HC})$ is the number of GPS track points whose direction change is greater than the threshold $c_{HC}$, and $\text{Trajectory length}$ is the distance of the travel segment.
Figure 11. Acceleration of different travel patterns with driving distance.

Figure 12. Absolute value curve of acceleration in different travel patterns.
VCR = \frac{N(VC \geq c_{vc})}{\text{Trajectory length}} \quad \text{VC} = |V1 - V2| \quad (11)

where VC represents the speed change, \( N(VC \geq c_{vc}) \) is the number of GPS track points whose speed change is greater than the threshold \( c_{vc} \), and \( \text{Trajectory length} \) is the distance of the travel segment.

Considering that the calculation formulas of the three eigenvalues of SR, HCR, and VCR need to determine the threshold, this article will first determine the optimal threshold, and then calculate the eigenvalue and construct the model. In order to determine the calculation threshold of the eigenvalue SR, the parameter \( c_{vc} \) in formula (11) needs to be adjusted. The test scheme is as follows: \( c_{vc} \in [0.5, 5], \text{step} = 0.5 \). When \( c_{vc} \) takes \( 2.5 \) m/s, the recognition result of the SVM model is optimal, so this article calculates the eigenvalue SR with \( 2.5 \) as the optimal parameter.

Similarly, the HCR calculation method is determined. The parameters were replaced by \( c_{HC} \) and \( c_{HC} \in [5, 30], \text{step} = 5 \). When \( c_{HC} \) takes \( 10 \), the recognition result of the SVM model is optimal, so this article calculates the eigenvalue HCR with \( 10 \) as the optimal parameter. After the feature value selection is completed, three machine learning models will be used to complete the travel pattern recognition work.

Travel pattern recognition based on machine learning model

SVM-based travel pattern recognition. The principle of SVM is given in Wang et al.\(^4\) The experimental scheme is designed as follows: two inputs are made. The first is the actual travel segment and the extracted feature value of the line segment, and the actual travel segment label, where the feature value is all the feature values mentioned in section “Feature value selection.” The second input is the same as the first, but the feature values \( \max|a1|, |a2|, |a3| \) are deleted. Then, SVM is constructed for classification, with 70% of the actual travel segment as the training set, 30% of the data as the test set, and the optimal parameters are selected to determine the optimal input. Among them, the adjustable parameters include the SVM kernel function \( \text{kernel} \), which ranges from \{“rbf,” “poly,” “sigmoid,” “linear”\}; SVM built-in parameter \( C \), the value range is \([1,10,100]\); built-in parameters \( \text{gamma} \), the range is \([1e^{-1}, 1e^{-2}, 1e^{-3}]\); and the \( \text{degree} \) in the kernel function, which ranges from \([1, 3, 5, 7, 9]\). The evaluation index is F1-score, and the output of the model is the classification result of the extracted line segment, the model evaluation index, and the operation time.

When adding the eigenvalues \( \max|a1|, |a2|, |a3| \) and not added, the model optimal parameter combination is \( \text{kernel} = \text{“rbf,”} \ C = 10, \text{ and } \text{gamma} = 1e^{-3} \). Precision rate, recall rate, and F1-score are shown in Tables 19 and 20. It can be seen from the results of
Tables 19 and 20 that although in the above-mentioned travel pattern characteristic analysis part, the discrimination effect of the acceleration-related feature value is not obvious, but in the SVM model, the feature value still plays a certain role. Therefore, these feature values are retained in this article.

Travel pattern recognition based on decision tree. The principle of decision trees is given in Bai.19 Over-fitting problems are more common because decision trees may build overly complex models to approximate the distribution of training sets. Usually, the problem is solved by pruning, setting the minimum number of sample nodes for the leaf nodes, and setting the maximum depth of the decision tree. Therefore, the decision tree model test scheme is set as follows: the input is the actual travel segments, the extracted feature values (the feature values are all the feature values mentioned in section “Feature value selection”), and the actual travel segment labels. Take 70% of the actual travel segments as the training set, 30% of the data as the test set, construct the decision tree model for classification, and select the optimal parameters. The adjustable parameters include the maximum depth of the decision tree (max_depth) and the minimum number of samples of the child node (min_sample_leaf). The evaluation index is F1-score, and the model output is the classification result of the extracted line segment, the model evaluation index, and the operation time. When max_depth = 9 and min_sample_leaf = 10, the model score is the highest. The decision tree model identification results under this parameter are shown in Table 21.

| Table 19. SVM optimal parameter identification results when adding eigenvalues max |a1|, |a2|, |a3|. | Table 20. SVM optimal parameter identification results when not adding eigenvalues max |a1|, |a2|, |a3|. |
|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Support | Precision | Recall | F1-score | Support |
| Walk | 0.87 | 0.92 | 0.90 | 864 | Walk | 0.84 | 0.91 | 0.87 | 864 |
| Bike | 0.83 | 0.83 | 0.83 | 369 | Bike | 0.81 | 0.73 | 0.77 | 369 |
| Bus | 0.82 | 0.75 | 0.78 | 450 | Bus | 0.74 | 0.49 | 0.59 | 424 |
| Car | 0.77 | 0.72 | 0.75 | 213 | Car | 0.52 | 0.72 | 0.61 | 221 |
| Average | 0.84 | 0.84 | 0.84 | 1896 | Average | 0.79 | 0.78 | 0.78 | 1896 |

Travel pattern recognition based on CNN

Overview of the principle of CNNs. CNN is a widely used deep neural network.5 It has unique advantages in processing topological data. It can process both one-dimensional time series data and two-dimensional image data. The input of the CNN is a fixed-size multi-dimensional tensor. The model mainly has three types of layers: convolutional layer, pooled layer, and fully connected layer. The overall structure is shown in Figure 14.

The convolutional layer consists of a set of learnable filters. By continuously sliding, the filter can learn the local information features of the image. This way, the information can be connected. Each neuron in the convolutional layer output is connected to a small area (receiving field) of the previous layer, where the size of the receiving field is consistent with the filter. The output value of the neuron is calculated by operating the dot product between the parameters of the filter and the entrance of the receiving field. Rotating the same filter over the entire surface of the input body creates a two-dimensional map in the output body, also known as a feature map or activation map. Performing a similar operation on all layers of filters creates several activation maps. Superimposing the feature map of all filters along the depth direction creates a three-dimensional output volume for that layer. The three-dimensional output shape of each convolutional layer is controlled by three hyperparameters specified by the user: the first is the depth of the output volume, which corresponds to the number of filters used for the convolution operation; the second is the step size, which is the number of elements that the filter moves on the input volume at a time; and the third is zero padding, which is the process of adding zero values at the beginning and end of the input volume matrix to control the amount of space in the output.

For travel data, it needs to be converted into approximate image matrix data, and an m × n matrix is formed by the number n of features and the number m of track points in the travel segment. Since CNN requires the input shape of the data to be consistent, the number of track points in different travel segments is usually different. Therefore, this article limits the
number of track points for each travel segment to 300 (the median of the number of track points for all track segments) and divides the travel segments for more than 300 track points into multiple input matrices for less than 300 tracks. The travel segment of the point is implemented with a zero-compensation strategy. The input matrix based on features and track points can be expressed by equation (12). Each input matrix uniquely corresponds to a tag matrix, and the tag matrix stores the actual travel tags corresponding to the travel segment

$\text{Input} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$ (12)

where $n = 5$ represents the five eigenvalues: $v$, $a$, $H$, $HC$, and $VC$. $x_g$ represents the value of the $i$th GPS track point under the $i$th eigenvalue.

The purpose of the pooling layer is to achieve spatial and scale invariance, reduce computational complexity, and control overfitting by reducing the dimensionality of each feature map and spatially downsampling the convolutional layer. The largest pooling layer is the most common type of pooling layer, which divides each depth slice of the input volume into non-overlapping vectors, and then selects the maximum value in each sub-vector as the representative value of the vector. The filter size of the largest pooled layer determines the length of the sub-vector (i.e. the number of entries the maximum value should be taken over). Therefore, the depth of the input and output volumes in the pooled layer is the same.

The fully connected layer is the end of the CNN network. Like traditional multi-layer perceptron, each neuron in the fully connected layer is connected to all neurons in the previous layer and is multiplied by unit direction. Except for the last layer, the other layers play the role of extracting features. Subsequently, the advanced features extracted from the first few layers are input to the last layer to perform the classification task, where the softmax activation function is used to generate a probability distribution on the label. The model structure can be adjusted and optimized by increasing and decreasing the number of fully connected layers and the number of neurons per layer. The model finally corrects the weight through backpropagation and finally completes the training process.

**Experimental plan and results.** In order to adjust the network structure to achieve better experimental results, this article has set up several CNN structures as shown in Table 22. The first parameter of each Conv2D represents the number of filters, the second parameter represents the size of each kernel, and each MaxPooling2D parameter represents the size of pooling windows.

Comparing models 1, 2, and 3, when the convolution and pooling layer networks are deepened and the number of convolution kernels is increasing, but the full connection layer is not added, the CNN first achieves a superior performance level. However, as the complexity of the model increases, the performance level of the model also declines, which may be caused by the amount of data. Comparing models 1, 4, and 5, when adding the fully connected layer, the performance of the model can be improved, but the number of fully connected layers should not be set too much. Therefore, the model 4 is the optimal model (as shown with bold values in Table 22). The model 4 training results are shown in Figure 15, and the classification results are shown in Table 23.

**Comparison and analysis of experimental results.** Comprehensive comparison of three machine learning models, the recognition results under the optimal parameters are as follows: SVM operation takes 26 s, F1-score is
0.84; decision tree operation takes 8 s, F1-score is 0.78; and CNN takes 384 s, F1-score is 0.80. Among the three models, SVM performs best on this problem, while decision tree and CNN perform quite well. The main reason may be that the data volume is small, but the data complexity is relatively high. For decision trees, the model cannot fully learn the connections and rules within the data while preventing overfitting. For the CNN, the complex model structure can better mine the characteristics of the data, but the current data volume is too small for the CNN, and the model cannot be fully trained. If the amount of data can be increased, the performance of the CNN on this issue is still worth looking forward to.

From the perspective of computing time, the decision tree takes the shortest time and the CNN is the longest. SVM achieves a perfect balance of F1-score and computing time under the current data volume. However, if the amount of data is increased, whether the SVM operation time will increase sharply, whether the F1-score performance will be surpassed by the CNN remains to be further verified.

Finally, this article selects the optimal SVM model to classify the extracted travel pattern. Since the transition point recognition is not completely correct, the actual travel pattern corresponding to the extracted line segment may contain more than one. Therefore, this article evaluates the travel pattern recognition result by comparing the correct proportion of GPS track point labels. By calculating the ratio of the track points that match the true results to the total number of track points, the accuracy is 76.83%.

### Summary and outlook

Travel pattern recognition based on GPS data can describe user travel rules and features conveniently, efficiently, and accurately. The research results not only can improve the convenience and efficiency of users' travel but also bring the possibility of profitability to digital map service providers. It will also assist the optimization and upgrading of urban traffic structure and benefit many parties.

This article first summarizes the research status of the field, and on this basis, selects the GeoLife GPS Trajectories public data set released by the Microsoft Research Asia in 2012 as experimental data, and identifies four travel patterns: walking, bicycle, bus, and taxi/private car.

This article mainly completes the trajectory segmentation through the transformation point recognition technology. By counting the data set labels, the experiment found that most of the transfer will have a walking segment, so the start point and the end point of the walking segment have a higher probability of becoming a transition point. In the process of distinguishing between walking and non-walking points, this article proposes a classification method that relates the density-based characteristics of DBSCAN clustering to the density distribution of GPS trajectory points, and obtains a better classification in walking and non-
walking point recognition. As a result, it laid a solid foundation for subsequent experiments.

After that, by combining and correcting the travel segments, this article proposes a series of eigenvalues to effectively distinguish different travel patterns, and constructs SVM model, decision tree model, and CNN to compare the performance of different models. The results show that the SVM performs best on this problem, and the F1-score reaches 0.84 under the optimal parameters.

Although this article proposes some improvement schemes for conversion point recognition, there are still many imperfections in the research: for the GPS trajectory data, a more optimized noise reduction processing algorithm is still not found. At the same time, the calculation of the characteristics such as instantaneous velocity is low due to the low sampling frequency of the data, which tends to cause large deviation of the calculation results and affect the final prediction result.

As for the prospect of the work of this article, because the classification accuracy rate is used as the evaluation index in the experiment process, the optimal parameters of DBSCAN clustering are selected by the homogeneity score. The adjustment process needs to be compared according to the actual label, and then selects the optimal value. Therefore, in the future, a parameter adaptive method can be adopted. On one hand, it can be better divided according to different trajectories and individual characteristics of different travelers, and on the other hand, parameter optimization can be realized more efficiently and conveniently.42

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

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by the National Natural Science Foundation of China (grant no. 61104166).

ORCID iDs
Juan Chen https://orcid.org/0000-0003-0631-8370
Shiyu Zhu https://orcid.org/0000-0001-7484-8459

References
1. Liu L and Duan PP. Research and application of smart city transportation network system. Comput Tech Develop 2019; 29(8): 185–189.
2. Zhang SH. Development of traffic big data and exploration of ideas. Public Commun Sci Tech 2019(2): 175–176.
3. Yang JF. Research on travel pattern recognition based on trajectory information. Beijing, China: North China University of Technology, 2018.
4. Zheng Y, Liu L, Wang L, et al. Learning transportation mode from raw GPS data for geographic applications on the Web. In: Proceedings of the 17th international conference on World Wide Web, Beijing, China, 21–25 April 2008, pp.247–256. New York: ACM Press.
5. Dabiri S and Heaslip K. Inferring transportation modes from GPS trajectories using a convolutional neural network. Transp Res Part C: Emerg Tech 2018; 86: 360–371.
6. Xiao G, Juan Z and Zhang C. Travel mode detection based on GPS track data and Bayesian networks. Comput Environ Urban Syst 2015; 54: 14–22.
7. Yang X, Stewart K, Tang L, et al. A review of GPS trajectories classification based on transportation mode. Sensors 2018; 18(11): 1–20.
8. Zhang B. Research on simplification and semantic enhancement of GPS space-time trajectory data for traffic trip survey. Shanghai, China: East China Normal University, 2011.
9. Resheff YS. Online trajectory segmentation and summary with applications to visualization and retrieval. In: 2016 IEEE international conference on big data (Big Data), Washington, DC, 5–8 December 2016. New York: IEEE.

10. Chang Y and Newman MW. Understanding how trace segmentation impacts transportation mode detection. In: 2012 ACM conference on ubiquitous computing, Pittsburgh, PA, 5–8 September 2012. New York: ACM.

11. Zhou Y. Research on individual travel mode recognition based on smartphone. Nanjing, China: Southeast University, 2017.

12. Zheng Y, Chen Y, Li Q, et al. Understanding transportation modes based on GPS data for web applications. ACM Trans Web 2010; 4(1): 58374.

13. Wang X. Research on travel-activity recognition method based on GPS data. Changchun, China: Jilin University, 2015.

14. Shen Y. Research on traffic travel mode recognition based on GPS trajectory data. Beijing, China: Beijing Jiaotong University, 2017.

15. Li YR, Chen HM, Wang LZ, et al. Density-based retention point identification method. Big Data Research 2018; 4(5): 80–93.

16. Xiang LG and Shao XT. Kernel density method for vector trajectory retention information extraction and its visualization. Acta Geodaetica Et Cartograp Sinica 2016; 45(9): 1122–1131.

17. Huang G, He J, Zhou W, et al. Discovery of stop regions for understanding repeat travel behaviors of moving objects. J Comput Syst Sci 2016; 82: 582–593.

18. Zhou Y and Yang C. Research on trajectory stop point recognition based on spatio-temporal clustering algorithm. J Transp Syst Eng Inform Tech 2018; 18(4): 88–95.

19. Bai Y. Research on travel mode identification method based on GPS data. Changchun, China: Jilin University, 2016.

20. Zheng Y, Zhang LZ, Xie X, et al. Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th international conference on World Wide Web, Madrid, 20–24 April 2009, pp.791–800. New York: ACM.

21. Zheng Y, Xie X and Ma WY. GeoLife: a collaborative social networking service among user, location and trajectory. IEEE Data Eng Bullet 2010; 32-40: 1–7.

22. Stenneth L, Wolfson O, Yu PS, et al. Transportation mode detection using mobile phones and GIS information. In: 19th ACM SIGSPATIAL international conference on advances in geographic information systems, Chicago, IL, 1–4 November 2011. New York: ACM.

23. Ye SN. Traffic pattern recognition based on deep convolution network. Chengdu, China: Southwest Jiaotong University, 2018.

24. Liu G, Yun M, Tan G, et al. Identification method of transportation mode based on GPS data. In: American society of civil engineers 15th COTA international conference of transportation professionals, Beijing, China, 25–27 July 2015. Beijing, China: Beijing Jiaotong University.

25. Zheng Y, Li Q, Chen Y, et al. Understanding mobility based on GPS data. In: 10th international conference on ubiquitous computing, Seoul, 21–24 September 2008. New York: ACM.

26. Witayangkurn A, Horanont T, Ono N, et al. Trip reconstruction and transportation mode extraction on low data rate GPS data from mobile phone. In: Proceedings of the international urban planning and urban management computer conference, Utrecht, 2–5 July 2013, pp.1–9. CUPUM.

27. Yao ZX. Research on information collection method of individual travel chain based on mobile phone sensor data. Chengdu, China: Southwest Jiaotong University, 2018.

28. Kasahara H, Liyama M and Minoh M. Transportation mode inference using environmental constraints. In: 11th international conference on ubiquitous information management and communication, Beppu, Japan, 5–7 January 2017. New York: ACM.

29. Chen JX, Zhao HL, Li RZ, et al. Travel mode recognition method based on mobile Internet technology. Comput Eng Design 2015; 36(9): 2532–2538.

30. Wang L, Zuo ZY and Fu JH. Research on feature analysis and recognition of travel mode based on SVM. J Transp Syst Eng Inform Tech 2014; 14(3): 70–74.

31. Li J, Sun J and Mi XY. Research on traffic travel mode recognition based on big data of smartphone. Appl Res Comput 2016; 33(12): 3527–3529, 3558.

32. Zhang ZH. Research on travel information extraction based on GPS trajectory. Shanghai, China: East China Normal University, 2010.

33. Matsuo N and Takaki K. Algorithm for capturing action history from GPS log information. In: IET 3rd international conference on wireless, mobile and multimedia networks, Beijing, China, 26–29 September 2010. New York: IEEE.

34. Wang L, Zuo ZY, Cao C, et al. Trip activity chain pattern recognition and travel trajectory data mining. In: 5th international conference on transportation engineering, Dalian, China, 26–27 September 2015, pp.2026–2034. Reston, VA: ASME.

35. Etemad M, Soares A and Hoseyni A. A trajectory segmentation algorithm based on interpolation-based change detection strategies. CEUR Workshop 2019, http://ceur-ws.org/Vol-2322/BMDA_4.pdf

36. Ghazi I, Maqbool MR, Haq IU, et al. GPS based autonomous vehicle navigation and control system. In: 13th international Blurban conference on applied sciences and technology, Islamabad, Pakistan, 12–16 January 2016, pp.238–244. New York: IEEE.

37. Ariffin AAB, Aziz NHA and Othman KA. Implementation of GPS for location tracking. In: 2011 IEEE control and system graduate research colloquium, Shah Alam, Malaysia, 27–28 June 2011, pp.77–81. IEEE.

38. Yang B. Research on recognition of college students’ travel mode based on smart phone GPS. Zhenjiang, China: Jiangsu University, 2017.

39. Rosenberg A and Hirschberg J. V-measure: a conditional entropy-based external cluster evaluation measure. In: 2007 joint conference on empirical methods in natural language processing and computational natural language learning, Prague, 28–30 June 2007, pp.410–420. Stroudsburg, PA: Association for Computational Linguistics.
40. Grira N, Crucianu M and Boujemaa N. Unsupervised and semi-supervised clustering: a brief survey, 2005, http://cedric.cnam.fr/~crucianm/src/BriefSurveyClustering.pdf

41. Wang HY, Li JH and Yang FL. A survey of support vector machine theory and algorithms. Appl Res Comput 2014(5): 1281–1286.

42. Li WJ, Yan SQ, Jiang Y, et al. Algorithm research on adaptive determination of DBSCAN algorithm parameters. Comput Eng Appl 2019; 55: 1–7.

43. Li G, Chen C, Huang S, et al. Public transportation mode detection from cellular data. In: International Conference on Information and Knowledge Management, Singapore, 6–10 November 2017, pp.2499–2502. New York: ACM.