

Research Article

Transportation Mode Detection Based on Permutation Entropy and Extreme Learning Machine

Lei Zhang,¹ LeiJun Liu,¹ SuNing Bao,¹ MengTing Qiang,¹ and XiaoMei Zou²

¹*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China*

²*School of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*

Correspondence should be addressed to Lei Zhang; zhanglei@cumt.edu.cn

Received 14 August 2015; Revised 3 October 2015; Accepted 8 October 2015

Academic Editor: Michael Small

Copyright © 2015 Lei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the increasing prevalence of GPS devices and mobile phones, transportation mode detection based on GPS data has been a hot topic in GPS trajectory data analysis. Transportation modes such as walking, driving, bus, and taxi denote an important characteristic of the mobile user. Longitude, latitude, speed, acceleration, and direction are usually used as features in transportation mode detection. In this paper, first, we explore the possibility of using Permutation Entropy (PE) of speed, a measure of complexity and uncertainty of GPS trajectory segment, as a feature for transportation mode detection. Second, we employ Extreme Learning Machine (ELM) to distinguish GPS trajectory segments of different transportation. Finally, to evaluate the performance of the proposed method, we make experiments on GeoLife dataset. Experiments results show that we can get more than 50% accuracy when only using PE as a feature to characterize trajectory sequence. PE can indeed be effectively used to detect transportation mode from GPS trajectory. The proposed method has much better accuracy and faster running time than the methods based on the other features and SVM classifier.

1. Introduction

With the increasing prevalence of positioning technologies, GPS mobile devices, smartphones, and so forth are equipped with multiple sensors [1]. It is possible to collect movement data of human. This makes it possible to implement various location-aware services [2]. Humans travel by different transportation modes, for example, walking, bicycle, car, and train [3]. In ubiquitous and context aware computing, understanding the mobility of a mobile user is an important research area. The knowledge of the transportation mode is critical for travel behavior research, transport planning, and traffic management [4, 5]. Transportation modes of individuals effectively reflect their past events and we can deeply understand their own life pattern [6]. Because the collected data from GPS do not contain the transportation mode, the detection of transportation mode from GPS trajectory is necessary.

Transportation mode detection from GPS data has been studied in the literature. Different studies use different features (or combination of features), such as speed, acceleration, maximum or median speed, and acceleration and

length between GPS fixes. A simple approach is to measure the speed and acceleration of the GPS data, which is then compared with empirical thresholds [5, 7]. However, for some transport modes, such as cycling and running, the usage of speed and acceleration is not enough. For example, for traffic jam, rain, and snow weather, the speed and acceleration under different transportation modes may be the same, so it is hard to differentiate using only the speed and acceleration thresholds. Zheng et al. identified a set of sophisticated features including heading change rate, velocity change rate, and stop rate [6, 8]. Beyond simple velocity and acceleration, they are more robust to traffic condition and contain more information of users' motion. Stenneth et al. considered transportation network data which consist of real time locations of buses, rail lines, and bus stops spatial data [9]. This approach can achieve over 93.5% accuracy for inferring various transportation modes. However, transportation network data is not available in most cases. Reddy et al. combined GPS sensor data with accelerometer data to detect the modes of transportation [10]. They select GPS speed, accelerometer variance, and accelerometer DFT as features.

Stopher et al. considered the average speeds and the maximum and minimum speed as feature set [11]. The other features used in transportation mode detection are shown in [12, 13].

Permutation Entropy (PE) directly investigates the temporal information contained in the time series, which was introduced by Bandt and Pompe [14]. PE has the quality of simplicity, robustness, and very low computational cost. PE has been applied in many applications [15], such as neural applications [16], electroencephalography (EEG) signal analysis [17], electrocardiograph (ECG) [18], and stock market analysis [19]. As a new learning algorithm for single-hidden layer feedforward neural networks, Extreme Learning Machine (ELM) has attracted a lot of research interests [20–23]. ELM has shown good performance in classification applications due to the low computational cost, better generalization performance, and faster learning speed than traditional gradient-based learning algorithms [24, 25].

PE has not been used for analyzing moving objects data. Therefore, it is interesting to investigate PE in mobility analysis of moving object. In this paper, we propose to use PE as a feature for the transportation mode detection. To reduce the training time without compromising accuracy, Extreme Learning Machine (ELM) is used as a classifier in this paper. Experiments are conducted on GeoLife dataset to validate the feasibility of the proposed method and evaluate the effectiveness of this approach. We make comprehensive performance evaluation for various feature measures as well as different supervised classifiers in transportation mode detection. The results showed that the proposed scheme was capable of detecting transportation not only with high accuracy but also with a very fast speed.

The remainder of this paper is organized as follows. In Section 2, we introduce the proposed transportation modes detection algorithm based on PE and ELM. In Section 3, we review the concept of PE. In Section 4, we provide a review of ELM. In Section 5, we present experiments and results to demonstrate the effectiveness of the algorithm. Section 5 contains conclusions.

2. Transportation Mode Detection with PE and ELM

In transportation modes detection from GPS data, different features, such as speed, average speed, acceleration, and sophisticated features, are used. But the speed of different transportation modes is usually vulnerable to traffic conditions and weather. Intuitively, the average speed of driving would be as slow as walking in congestion. Speed change of GPS trajectories is an important indicator to describe trajectories. For example, the speed of car changes in a wide range. Compared with this, the speed of walking has less change. Permutation Entropy estimates the complexity of time series through the comparison of neighboring values. It is conceptually simple and computationally very fast. So, we explore whether Permutation Entropy can be used to detect transportation modes.

In the paper, we adopt Permutation Entropy as a measure of complexity due to its fast calculation, robustness, and

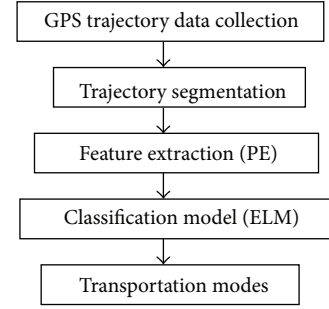


FIGURE 1: The steps of transportation mode detection method based on PE and ELM.

invariance with respect to nonlinear monotonous transformations.

Up to now, to the best of our knowledge, in the literature, there is no related study about detecting moving objects' transportation mode by PE. Figure 1 shows the proposed transportation mode detection method with PE and ELM.

Firstly, GPS trajectory data are collected from GPS sensor. Secondly, trajectory data is segmented into time sequences with the same length. For each trajectory segment, several features, such as speed and PE, are extracted. Finally, a classification model, ELM, is used to detect transportation mode.

3. Permutation Entropy (PE)

Permutation Entropy is widely used to study the irregularity and nonlinearity in time series, which has a fairly high sensitivity on time, so it is an effective method to detect the dynamic changes of a complex system.

For a given time series $\{x(i)\}_{i=1}^N$, the calculation steps of PE are described as follows.

(1) Using time delay embedding theorem to reconstruct the phase space, the data segment $X(i)$ is derived from the point $x(i)$ of original time series:

$$X(i) = [(x(i), x(i + \tau), \dots, x(i + (m - 1)\tau))]. \quad (1)$$

In (1), m is the embedding dimension and τ is the delay time.

(2) Each component in $X(i)$ can be arranged in an increasing order to achieve an ordinal pattern:

$$[x(i + (j_1 - 1)\tau) \leq x(i + (j_2 - 1)\tau) \leq \dots \leq x(i + (j_m - 1)\tau)], \quad (2)$$

where j_* is the index of the element in the new vector.

(3) If there are two same values in $X(i)$, for example, $x(i + (j_{i1} - 1)\tau) = x(i + (j_{i2} - 1)\tau)$, we rearrange them according to the index. Namely, if $j_{i1} \leq j_{i2}$, then, $x(i + (j_{i1} - 1)\tau) \leq x(i + (j_{i2} - 1)\tau)$.

(4) Each vector $X(i)$ can be mapped into a symbol series $A(i) = (j_1, j_2, \dots, j_m)$. $A(i)$ is one of the $m!$ permutations of m distinct symbols. The probability distribution for each distinct symbol series can be estimated, P_1, P_2, \dots, P_K , where $K \leq m!$.

(5) PE for $\{x(i)\}_{i=1}^N$ is defined as the Shannon entropy for K different symbols:

$$H_p(m) = -\sum_{j=1}^K P_j \ln P_j. \quad (3)$$

When $P_j = 1/m!$, we can get the maximum value of $H_p(m)$, $H_p(m) = \ln(m!)$.

4. Extreme Learning Machine

4.1. Single-Hidden Layer Feedforward Neural Network (SFLN).

Feedforward neural networks have been extensively used in many fields. A single-hidden layer feedforward neural network (SLFN) with at most M hidden nodes and with almost any nonlinear activation function can exactly learn M distinct observations. The activation function of a node defines the output of that node given an input or set of inputs. The input weights (linking the input layer to the first hidden layer) and hidden layer biases need to be adjusted.

For M arbitrary distinct samples (x_i, y_i) , where $x_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T \in R^n$ and $y_i = [y_{i1}, y_{i2}, \dots, y_{im}]^T \in R^m$, SLFN with N hidden nodes and activation function $g(x)$ is modeled as

$$\sum_{i=1}^N \beta_i \cdot g(w_i \cdot x_j + b_i) = o_i, \quad j \in [1, M], \quad (4)$$

where $w_i = [w_{i1}, w_{i2}, \dots, w_{iN}]^T$ is the weight vectors connecting the input nodes to the i th hidden node, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ is the weight vectors connecting i th hidden node to the output nodes, and b_i is the threshold of the i th hidden node. $w_i \cdot x_j$ denotes the inner product of w_i and x_j .

The standard SLFN with N hidden neurons can approximate these N samples with zero error; that is, $\sum_{j=1}^N \|o_j - t_j\| = 0$, and there exist $\{\beta_i, w_i, b_i\}_{i=1}^M$ such that

$$\sum_{i=1}^M \beta_i g(w_i^T x_j + b_j) = t_j, \quad j = 1, \dots, N. \quad (5)$$

The above M equations in (4) can be written compactly as

$$H\beta = Y, \quad (6)$$

where H is defined as

$$H = \begin{bmatrix} f(w_1 x_1 + b_1) & \cdots & f(w_N x_1 + b_N) \\ \vdots & \ddots & \vdots \\ f(w_1 x_M + b_1) & \cdots & f(w_N x_M + b_N) \end{bmatrix}, \quad (7)$$

$$\beta = [\beta_1, \beta_2, \dots, \beta_N]^T,$$

$$Y = [y_1, y_2, \dots, y_M]^T.$$

H is called the hidden layer output matrix of the neural network; the i th column of H is the i th hidden node output with respect to inputs x_1, x_2, \dots, x_M .

Traditionally, all the parameters of the feedforward networks need to be tuned iteratively. Gradient descent-based methods have mainly been used in various learning algorithms of feedforward neural networks. Gradient descent-based learning methods are generally very slow due to improper learning steps or may easily converge to local minima.

4.2. Extreme Learning Machine. Extreme Learning Machine (ELM) is a simple learning algorithm for SFLN. The learning speed of ELM can be thousands of times faster than traditional feedforward network learning algorithms while obtaining better generalization performance.

In most applications, the number of hidden neurons is much smaller than the number of distinct training samples, and H is a nonsquare matrix. In the ELM approach, the input weights w_i and the hidden layer biases b_i of SLFNs are not tuned but are assigned randomly and then fixed. This is equivalent to mapping the samples to a random feature space. Then, training SLFN is equivalent to find a least squares error solution $\hat{\beta}$ of the linear system $H\beta = Y$. $\hat{\beta} = H^*Y$ is the Moore-Penrose generalized inverse of matrix H .

There are many ways of calculating the Moore-Penrose generalized inverse of a matrix such as the orthogonal projection method, iterative method, and singular value decomposition [26]. Singular value decomposition is used to calculate the Moore-Penrose generalized inverse of a matrix.

5. Experiments Evaluation

In this section, we first describe experiment dataset. Second, we present feature extraction. Finally, transportation modes detection based on elementary features, only PE, and combination of PE and the elementary features is discussed.

We do not compare our method with the previous transportation modes detection methods because of the following. Firstly, the transportation mode detection method is composed of trajectory partition, feature selection, and learning process. Subtrajectories are attained automatically from trajectories partition algorithm in the other transportation mode detection methods. The length of subtrajectories is not the same. However, subtrajectories in our method need to have the same length to calculate PE. So, we get the fixed length subtrajectories by partitioning the trajectories with the same length. Other researchers partition the trajectories with the specific trajectories partition algorithms. Secondly, we compare PE with the other features in the other method to show that PE is a valid indicator. Finally, we compare our learning method, ELM, with SVM, commonly used learning method in the other methods, to validate ELM's efficiency.

5.1. Dataset Description. The experiments are carried out on the Microsoft GeoLife dataset [1] which consists of 17621 moving trajectories of 182 users over three years. These trajectories were recorded by different GPS loggers and GPS phones. A GPS trajectory is represented by a sequence of time-stamped points of a user in a certain time interval and each time-stamped point contains the information of latitude, longitude, and altitude. The trajectories of 73 users have been

TABLE 1: Total distance and duration of transportation modes.

Transportation mode	Distance (km)	Duration (hour)
Walk	10123	5460
Bike	6495	2410
Bus	20281	1507
Car and taxi	32866	2384
Train	36253	745
Airplane	24789	40
Other	9493	404
Total	140304	12953

TABLE 2: Extracted features of each trajectory segment.

Category	Features	Significance
Basic features	AV	Average velocity
	DV	Velocity variance
Advanced features	HCR	Heading change rate
	SR	Stop rate
	VCR	Velocity change rate
	PE	PE of velocity

TABLE 3: Different training set sizes and the used features.

Sample sizes	Training	10%	20%	30%	40%	50%
	Testing	90%	80%	70%	60%	50%
Features	AV, DV, HCR, SR, and VCR					

labeled with transportation mode. The total distance and duration of transportation modes are listed in Table 1.

5.2. Feature Extraction. We extract the features from each trajectory. The features are shown in Table 2. The elementary features {AV, DV, HCR, SR, and VCR} have the same definition in [9].

5.3. Experimental Results

5.3.1. Transportation Modes Detection Based on Elementary Features. We select 30 of 73 users and extract 5525 trajectories to perform the experiments. The elementary features {AV, DV, HCR, SR, and VCR} are calculated to detect trajectory modes. 5525 trajectory segments are partitioned into the training set and the testing set randomly. Table 3 lists different training set sizes and the used features.

We choose SVM and ELM as classifiers to detect transportation modes. Tables 4–8 show the running time and accuracy. We observe that classification accuracy of ELM is about 62% when nodes number is larger than 500. We can get higher and steady results when nodes number of ELM is 800. Detection accuracy of SVM is about 45% and lower than ELM. For ELM, different activation functions have great effect on running time and accuracy. Sigmoid can get the most accurate result. The accuracy of Hardlim is slightly less than Sigmoid. However, training time of Hardlim is much shorter than Sigmoid.

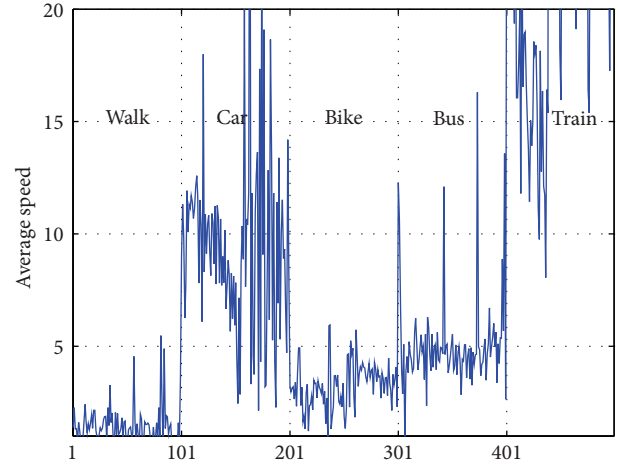


FIGURE 2: Average speed of 500 trajectory segments.

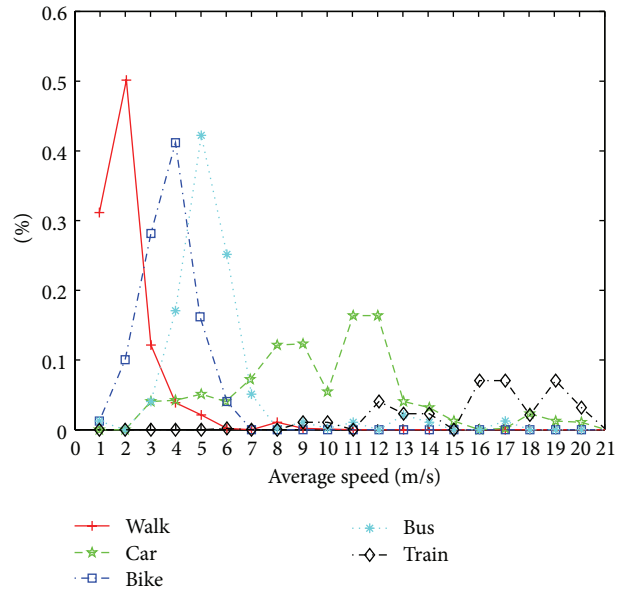


FIGURE 3: Speed distribution of different transportation modes.

5.3.2. Transportation Modes Detection Based on Only PE. To compute the speed PE of transportation modes, we extract trajectory segments with the same transportation mode. The number of points of each trajectory segment is greater than 1000. We collect 500 trajectory segments in our experiment. We calculate the speed of each point and the speed of transportation modes by using PE of each trajectory segment. We use the speed from PE as a feature to detect transportation modes.

Figure 2 shows the average speed of 500 trajectory segments. Figure 3 presents the speed distribution of different transportation modes. The range of speed change in each transportation mode is high. Different transportation modes, such as walk, bike, and bus, have high overlap in the average velocity. Consequently, the average velocity is not a perfect feature to distinguish different transportation modes.

TABLE 4: Running time and accuracy of 1 : 9 data.

1 : 9	Time (s)		Training accuracy	Testing accuracy	Node	Activation function
	Training	Testing				
ELM	0	0.0187	51.21%	44.47%	50	
ELM	0.0187	0.0374	56.53%	49.12%	100	
ELM	0.0655	0.1186	63.04%	52.39%	300	
ELM	0.1747	0.2028	69.66%	58.72%	500	Hardlim
ELM	0.5117	0.2995	77.76%	62.31%	800	
ELM	0.8268	0.3682	78.55%	60.61%	1000	
ELM	1.3229	0.4274	79.93%	58.87%	1200	
ELM	1.2542	0.5897	70.46%	62.53%	800	
ELM	1.9282	0.6614	82.10%	61.93%	1000	Sigmoid
ELM	2.8143	0.8018	82.75%	62.27%	1200	
SVM	0.0886	0.2261	—	39.984%	—	—

TABLE 5: Running time and accuracy of 2 : 8 data.

2 : 8	Time (s)		Training accuracy	Testing accuracy	Node	Activation function
	Training	Testing				
ELM	0.0125	0.0125	48.16%	46.42%	50	
ELM	0.0125	0.0312	58.28%	55.66%	100	
ELM	0.0842	0.0998	64.90%	57.81%	300	
ELM	0.2496	0.1716	71.26%	61.58%	500	Hardlim
ELM	0.6739	0.2621	74.79%	62.19%	800	
ELM	1.0265	0.3245	77.16%	64.06%	1000	
ELM	1.6068	0.3713	77.83%	62.70%	1200	
ELM	1.9438	0.4742	76.34%	64.28%	800	
ELM	3.1013	0.5959	77.94%	63.45%	1000	Sigmoid
ELM	4.5053	0.7582	80.16%	65.48%	1200	
SVM	0.1723	0.3869	—	40.068%	—	—

TABLE 6: Running time and accuracy of 3 : 7 data.

3 : 7	Time (s)		Training accuracy	Testing accuracy	Node	Activation function
	Training	Testing				
ELM	0.0094	0.0062	49.08%	48.78%	50	
ELM	0.0250	0.0312	55.25%	55.14%	100	
ELM	0.1279	0.0905	63.37%	58.70%	300	
ELM	0.3276	0.1435	68.02%	61.89%	500	Hardlim
ELM	0.8112	0.2402	71.87%	62.74%	800	
ELM	1.2636	0.2995	74.60%	63.42%	1000	
ELM	1.9563	0.3370	75.70%	63.03%	1200	
ELM	2.8704	0.4274	74.43%	64.70%	800	
ELM	4.2089	0.5554	75.54%	63.88%	1000	Sigmoid
ELM	6.4584	0.6521	78.85%	65.95%	1200	
SVM	0.3497	0.4839	—	40.652%	—	—

Figure 4 is the PE from the speed for different transportation modes. We can see that the PE from the speed in cars and buses is lower and PE from the speed in walking and bikes is higher. The PE from the speed in car and bus has smaller scale and lower value. The PE from speed in walking and bikes

has a larger scale and a higher value. When the speed of car, bus, walking, and bike is usual, we can recognize different transportation modes from the average speed easily. But, in the traffic jam, the average speed of car and bus is almost the same as the average speed of walking and bike. Because the PE

TABLE 7: Running time and accuracy of 4 : 6 data.

4 : 6	Time (s)		Training accuracy	Testing accuracy	Node	Activation function
	Training	Testing				
ELM	0.0125	0.0062	49.24%	47.59%	50	
ELM	0.0250	0.0343	55.93%	53.61%	100	
ELM	0.1654	0.0780	64.05%	58.02%	300	
ELM	0.4150	0.1466	69.79%	61.54%	500	Hardlim
ELM	1.0046	0.2153	73.05%	61.53%	800	
ELM	1.5850	0.2556	74.49%	62.97%	1000	
ELM	2.2683	0.2870	77.32%	63.87%	1200	
ELM	3.7192	0.3796	76.40%	65.56%	800	
ELM	5.7836	0.5252	77.74%	65.51%	1000	Sigmoid
ELM	8.1537	0.5564	78.31%	65.55%	1200	
SVM	0.6794	0.6037	—	40.513%	—	—

TABLE 8: Running time and accuracy of 5 : 5 data.

5 : 5	Time (s)		Training accuracy	Testing accuracy	Node	Activation function
	Training	Testing				
ELM	0.0229	0.0109	49.16%	49.22%	50	
ELM	0.0504	0.0268	54.90%	54.89%	100	
ELM	0.2454	0.0755	64.69%	62.39%	300	
ELM	0.5117	0.1309	68.09%	63.49%	500	Hardlim
ELM	1.1518	0.1908	71.87%	65.31%	800	
ELM	1.7061	0.2378	73.55%	65.61%	1000	
ELM	3.1928	0.2912	74.70%	65.87%	1200	
ELM	5.3716	0.4472	74.03%	67.90%	800	
ELM	8.0185	0.4628	75.74%	69.03%	1000	Sigmoid
ELM	11.6845	0.6240	77.55%	69.27%	1200	
SVM	2.382393	1.171633	—	42.816%	—	—

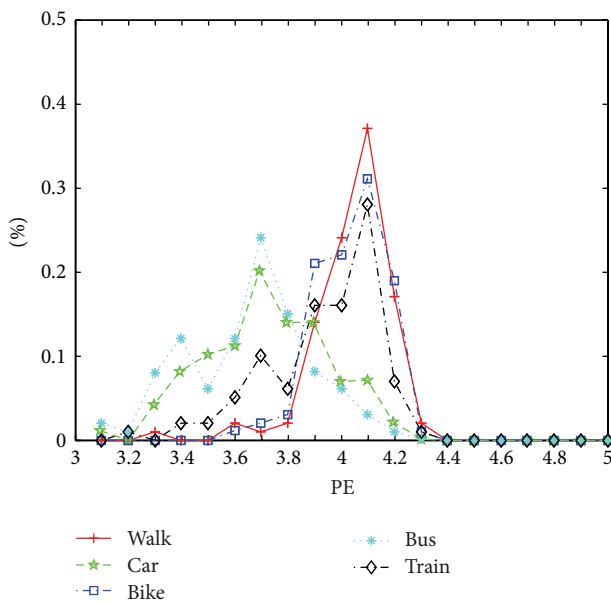


FIGURE 4: The distribution of velocity PE of different transportation modes.

TABLE 9: Average PE of 500 trajectory segments with different m under multiple transportation modes.

m	Transportation mode				
	Walk	Car	Bike	Bus	Train
$m = 5$	4.5775	4.0169	4.5908	4.0684	4.5276
$m = 6$	5.9285	5.1083	5.9421	5.1710	5.8707
$m = 7$	6.6051	5.8371	6.6218	5.8894	6.5289

from the speed for car is lower than that of walking, we can make a distinction between car and walking from PE from the speed.

The average PE of 500 trajectory segments with different m under multiple transportation modes is shown in Table 9. We can see that the average PE from the speed becomes large with the increase of m . This is possibly because when m is larger, the probability of distinct symbols is smaller and each row of the reconstruction matrix is much more complex.

We use the PE from the speed as the feature to detect transportation modes for 500 trajectory segments. We choose SVM and ELM as classifiers. In ELM, we adopt Sigmoid as

TABLE 10: Experimental results with $m = 4$.

Training data sizes	Classifier	Training time (s)	Testing time (s)	Training accuracy	Testing accuracy
10%	SVM	0.0089	0.0258	—	20.73%
	ELM	0.0033	0.0068	23.92%	22.35%
20%	SVM	0.0106	0.0129	—	22.16%
	ELM	0.0078	0.0020	22.11%	20.21%
30%	SVM	0.0115	0.0134	—	21.82%
	ELM	0.0041	0.0108	24.13%	24.88%
40%	SVM	0.0126	0.0138	—	22.50%
	ELM	0.0094	0.0053	19.36%	18.73%
50%	SVM	0.0131	0.0138	—	20.01%
	ELM	0.0054	0.0046	19.02%	18.06%

TABLE 11: Experimental results with $m = 5$.

Training data sizes	Classifier	Training time (s)	Testing time (s)	Training accuracy	Testing accuracy
10%	SVM	0.0107	0.0203	—	41.80%
	ELM	0.0031	0.0048	56.38%	54.72%
20%	SVM	0.0095	0.0142	—	43.56%
	ELM	0.0031	0.0022	55.67%	55.47%
30%	SVM	0.0167	0.0198	—	42.10%
	ELM	0.0039	0.0068	57.84%	55.42%
40%	SVM	0.0329	0.0150	—	43.16%
	ELM	0.0065	0.0037	60.77%	44.25%
50%	SVM	0.0366	0.0280	—	42.67%
	ELM	0.0013	0.0024	59.26%	54.19%

TABLE 12: Experimental results with $m = 6$.

Training data sizes	Classifier	Training time (s)	Testing time (s)	Training accuracy	Testing accuracy
10%	SVM	0.0018	0.0052	—	40.67%
	ELM	0.0016	0.0104	57.58%	50.84%
20%	SVM	0.0036	0.0074	—	44.78%
	ELM	0.0031	0.0090	56.44%	55.63%
30%	SVM	0.0056	0.0106	—	45.50%
	ELM	0.0031	0.0106	58.38%	55.93%
40%	SVM	0.0088	0.0102	—	41.62%
	ELM	0.0078	0.0062	61.47%	51.85%
50%	SVM	0.0167	0.0102	—	43.71%
	ELM	0.0104	0.0020	58.64%	53.91%

the activation function and the number of nodes is 800 since these parameters can give relatively good performance for ELM. To demonstrate the effect of the number of training samples, we design the experiments by setting different training set sizes (10%, 20%, 30%, 40%, and 50%) and the remaining samples act as the training set.

For m , Bandt and Pompe recommend $m = 3, \dots, 7$ [14, 15] and found that $m = 3$ and 4 may still be too small, and a value of $m = 5, 6$, or 7 seems to be the most suitable.

We set m as 4, 5, 6, and 7. Tables 10–13 present the experimental results. We find that when $m = 4$, it is too small to get better effect as shown in [15]. For $m = 5, 6$, or 7, we find

that when m is larger, the detection accuracy is lower and the training time is longer. At the same time, we note that the larger the dimension is, the more time PE computing needs. When $m = 5$, we obtain the best experimental results. For two kinds of classifiers, it is noted that ELM gives a better stability and a higher accuracy than SVM.

5.3.3. Transportation Modes Detection Based on PE and the Elementary Features. We gradually add the other elementary features based on PE. In ELM, we adopt Sigmoid as the activation function and the training data size is 50%. Tables 14–18 present different detection results with different feature sets.

TABLE 13: Experimental results with $m = 7$.

Training data sizes	Classifier	Training time (s)	Testing time (s)	Training accuracy	Testing accuracy
10%	SVM	0.0094	0.0121	—	40.32%
	ELM	0.0016	0.0296	58.59%	51.23%
20%	SVM	0.0036	0.0076	—	43.91%
	ELM	0.0125	0.0156	55.98%	54.60%
30%	SVM	0.0055	0.0091	—	44.50%
	ELM	0.0104	0.0094	56.59%	55.60%
40%	SVM	0.0082	0.0104	—	44.57%
	ELM	0.0140	0.0031	56.22%	55.40%
50%	SVM	0.0219	0.0190	—	44.73%
	ELM	0.0156	0.0094	57.08%	54.65%

TABLE 14: Detection results with PE + AV.

Classifier	Time (s)		Testing accuracy	Node
	Training	Testing		
ELM	0.0075	0.0031	83.65%	50
ELM	0.0085	0.0037	83.58%	80
ELM	0.0112	0.0045	83.56%	100
ELM	0.0274	0.0130	82.83%	200
ELM	0.0586	0.0181	82.30%	300
ELM	0.1716	0.0287	81.96%	500
ELM	0.4975	0.0459	81.77%	800
ELM	0.8168	0.0679	81.83%	1000
SVM	0.0161	0.0076	74.07%	—

TABLE 15: Detection results with PE + AV + HCR.

Classifier	Time (s)		Testing accuracy	Node
	Training	Testing		
ELM	0.0068	0.0017	83.54%	50
ELM	0.0076	0.0076	83.39%	80
ELM	0.0102	0.0097	83.33%	100
ELM	0.0293	0.0190	82.62%	200
ELM	0.0595	0.0217	82.33%	300
ELM	0.1613	0.0293	81.78%	500
ELM	0.5108	0.0412	81.42%	800
ELM	0.8296	0.0674	81.39%	1000
SVM	0.0159	0.0059	72.22%	—

We can obtain about 80% accuracy by PE + AV as features. It is obvious that the accuracy of our method has no obvious increase after adding HCR, SR, and VCR. When we use the features PE, AV, HCR, SR, VCR, and DV, the accuracy will decrease. This is partly because there is a negative correlation between different features.

6. Conclusions

In this paper, we have proposed a transportation mode detection method based on PE and ELM. We employ speed PE as the feature of trajectory segments. The low computational complexity of PE makes it become an excellent feature.

TABLE 16: Detection results with PE + AV + HCR + SR.

Classifier	Time (s)		Testing accuracy	Node
	Training	Testing		
ELM	0.0048	0.0037	83.85%	50
ELM	0.0086	0.0040	83.68%	80
ELM	0.0164	0.0057	83.69%	100
ELM	0.0295	0.0198	82.82%	200
ELM	0.0674	0.0244	82.37%	300
ELM	0.1699	0.0326	81.57%	500
ELM	0.5261	0.0435	81.38%	800
ELM	0.8933	0.0636	81.45%	1000
SVM	0.0146	0.0065	72.83%	—

TABLE 17: Detection results with PE + AV + HCR + SR + VCR.

Classifier	Time (s)		Testing accuracy	Node
	Training	Testing		
ELM	0.0062	0.0035	83.79%	50
ELM	0.0070	0.0048	83.70%	80
ELM	0.0150	0.0079	83.69%	100
ELM	0.0291	0.0150	83.12%	200
ELM	0.0646	0.0214	82.45%	300
ELM	0.1866	0.0448	81.70%	500
ELM	0.4908	0.0600	81.49%	800
ELM	0.7981	0.0799	81.60%	1000
SVM	0.0144	0.0074	72.01%	—

Experimental results based on the GeoLife dataset show that speed PE is a valid feature to detect transportation modes from trajectory segments and obtain more than 50% accuracy. We also apply ELM as a classifier and validate the notion that ELM performs faster and obtains a higher accuracy than SVM in our experiments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

TABLE 18: Detection results with PE + AV + HCR + SR + VCR + DV.

Classifier	Time (s)		Testing accuracy	Node
	Training	Testing		
ELM	0.0045	0.0036	73.85%	50
ELM	0.0109	0.0062	75.06%	80
ELM	0.0117	0.0097	74.01%	100
ELM	0.0337	0.0178	74.75%	200
ELM	0.0564	0.0256	73.64%	300
ELM	0.1774	0.0317	73.13%	500
ELM	0.4325	0.0537	71.79%	800
ELM	0.7972	0.0374	70.56%	1000
SVM	0.0345	0.0127	63.45%	—

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities (2014XT04).

References

- [1] K. Waga, A. Tabarcea, M. Chen, and P. Franti, "Detecting movement type by route segmentation and classification," in *Proceedings of the 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 508–513, Pittsburgh, Pa, USA, October 2012.
- [2] K. Waga, A. Tabarcea, and P. Franti, "Context aware recommendation of location-based data," in *Proceedings of the 15th International Conference on System Theory, Control, and Computing (ICSTCC '11)*, pp. 1–6, IEEE, Sinaia, Romania, October 2011.
- [3] J. P. Rodrigur, C. Comtois, and B. Slack, *The Geography of Transport Systems*, Routledge, New York, NY, USA, 2008.
- [4] F. Biljecki, H. Ledoux, and P. van Oosterom, "Transportation mode-based segmentation and classification of movement trajectories," *International Journal of Geographical Information Science*, vol. 27, no. 2, pp. 385–407, 2013.
- [5] W. Bohte and K. Maat, "Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: a large-scale application in the Netherlands," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 3, pp. 285–297, 2009.
- [6] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the web," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, New York, NY, USA, April 2008.
- [7] J. L. Wolf, M. G. S. Oliveira, P. Troped, C. E. Mathews, E. K. Cromley, and S. J. Melly, "Mode and activity identification using GPS and accelerometer data," in *Proceedings of the 85th Annual Meeting of the Transportation Research Board*, Washington, DC, USA, January 2006.
- [8] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08)*, pp. 312–321, ACM, Seoul, South Korea, September 2008.
- [9] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and GIS information," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11)*, pp. 54–63, ACM, Chicago, Ill, USA, November 2011.
- [10] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, article 13, 2010.
- [11] P. Stopher, E. Clifford, J. Zhang, and C. FitzGerald, *Deducing Mode and Purpose from GPS Data*, Institute of Transport and Logistics Studies, Sydney, Australia, 2008.
- [12] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, "Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification," *Computers, Environment and Urban Systems*, vol. 36, no. 6, pp. 526–537, 2012.
- [13] M. Garbe, C. Bünnig, A. Gutschmidt, and C. Cap, "Moving type detection without time information," in *Proceedings of the 6th IEEE International Conference on Semantic Computing (ICSC '12)*, pp. 318–324, Palermo, Italy, September 2012.
- [14] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical Review Letters*, vol. 88, no. 17, Article ID 174102, 2002.
- [15] Y. Cao, W.-W. Tung, J. B. Gao, V. A. Protopopescu, and L. M. Hively, "Detecting dynamical changes in time series using the per-mutation entropy," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 70, no. 4, Article ID 046217, 2004.
- [16] Z. Li, G. Ouyang, D. Li, and X. Li, "Characterization of the causality between spike trains with permutation conditional mutual information," *Physical Review E*, vol. 84, no. 2, Article ID 021929, 2011.
- [17] A. A. Bruzzo, B. Gesierich, M. Santi, C. A. Tassinari, N. Birbaumer, and G. Rubboli, "Permutation entropy to detect vigilance changes and preictal states from scalp EEG in epileptic patients: a preliminary study," *Neurological Sciences*, vol. 29, no. 1, pp. 3–9, 2008.
- [18] B. Graff, G. Graff, and A. Kaczkowska, "Entropy measures of heart rate variability for short ECG datasets in patients with congestive heart failure," *Acta Physica Polonica B, Proceedings Supplement*, vol. 5, p. 153, 2012.
- [19] L. Zunino, M. Zanin, B. M. Tabak, D. G. Pérez, and O. A. Rosso, "Forbidden patterns, permutation entropy and stock market inefficiency," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 14, pp. 2854–2864, 2009.
- [20] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1352–1357, 2009.
- [21] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [22] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran, "Online sequential fuzzy extreme learning machine for function approximation and classification problems," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 39, pp. 1067–1072, 2009.
- [23] Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2483–2490, 2011.

- [24] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16–18, pp. 3460–3468, 2008.
- [25] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.
- [26] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications*, John Wiley & Sons, 1972.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

