Semantic Trajectory Based Behavior Generation for Groups Identification

Yang Cao[1], Zhi Cai[1], Fei Xue[2], Tong Li[1], Zhiming Ding[1]
[1] Faculty of Information Technology, Beijing University of Technology
Beijing, 100124, China
[e-mail: caoyangcwz@emails.bjut.edu.cn; caiz@bjut.edu.cn; litong@bjut.edu.cn; zmding@bjut.edu.cn]
[2] College of Information, Beijing Wuzi University
Beijing, 101149, China
[e-mail: xuefei2004@126.com]
*Corresponding author: Zhiming Ding

Received April 2, 2018; revised May 1, 2018; accepted July 28, 2018; published December 31, 2018

Abstract

With the development of GPS and the popularity of mobile devices with positioning capability, collecting massive amounts of trajectory data is feasible and easy. The daily trajectories of moving objects convey a concise overview of their behaviors. Different social roles have different trajectory patterns. Therefore, we can identify users or groups based on similar trajectory patterns by mining implicit life patterns. However, most existing daily trajectories mining studies mainly focus on the spatial and temporal analysis of raw trajectory data but missing the essential semantic information or behaviors. In this paper, we propose a novel trajectory semantics calculation method to identify groups that have similar behaviors. In our model, we first propose a fast and efficient approach for stay regions extraction from daily trajectories, then generate semantic trajectories by enriching the stay regions with semantic labels. To measure the similarity between semantic trajectories, we design a semantic similarity measure model based on spatial and temporal similarity factor. Furthermore, a pruning strategy is proposed to lighten tedious calculations and comparisons. We have conducted extensive experiments on real trajectory dataset of Geolife project, and the experimental results show our proposed method is both effective and efficient.

Keywords: Mobility Data, Trajectory Semantic, Semantic Enrichment, Similar Behaviors, Groups Identification

1. Introduction

With the development of global positioning system (GPS) and the popularity of mobile devices with positioning capability (such as mobile phones, smart watches, and driving recorders) in Internet of things (IoT), collecting large-scale daily life trajectory data is feasible and easy. The value of trajectory data itself promotes the research of new fields. Location-based service (LBS) and intelligent transportation system (ITS) are the two most famous branches. A large number of applications and studies have been done in these fields, such as location-based advertising push [1], friends location sharing [2] in LBS and vehicle scheduling [3], urban traffic analysis [4] in ITS and so on.

Daily trajectories of moving objects (e.g., vehicles, people) record their activities in the real world, which reflect their lifestyle or behavior to some extent. Therefore, by analyzing the trajectory data, we can find out the life patterns or behavior patterns of moving objects and the correlation between them. It is of great significance for many applications. For example, in friend recommendation system, we can recommend friends according to same locations where users often stay; in the field of public security, for an emergency evacuation of large events, we can identify and cluster groups of similar behaviors based on their trajectories. Also, many studies use trajectories and intelligent optimization algorithms [5-11] for path planning in emergency evacuation. Therefore, how to identify groups of similar patterns based on trajectory data is a hot and hard research.

Unfortunately, most existing trajectory mining studies mainly focus on the temporal and spatial features of trajectories [12-17]. These conventional trajectory patterns do not have explicit semantic information and cannot express user's behaviors effectively. We believe such semantic information plays an important role in trajectory analysis. In particular, for two people that have a same social role, their life patterns will have similar semantic feature, even though their daily trajectories are different. However, so far, most studies of semantic trajectories can only make simple semantic annotations, and cannot make effective use of a large number of historical trajectories.

Trajectory data is a collection of GPS points, and we can get basic attributes easily, such as time, location, etc. According to those spatio-temporal characteristics of trajectories, semantic information can be presented to users. How to enrich trajectories with semantic features is an essential challenge. Some existing studies employ users or volunteers to label the semantic tags on their trajectories. Some other studies combine the points of interest (POIs) and the location to generate semantic trajectories. However, it is hard to find out stay regions quickly and to select POIs properly in tons of data.

Massive trajectories impose another challenge. The semantic similarity between two trajectories can be described by Jaccard similarity coefficient. However, moving objects usually have a large number of historical trajectories. It will be impossible to calculate their similarity by pairwise comparison of semantic trajectories. How to compress the historical trajectories as an abstract semantic trajectory is another important challenge for trajectory data mining.

To address the above challenges, this paper makes following contributions:

- This paper proposes a fast and efficient approach to get stay regions from daily trajectories based on a spatial and temporal context, then generate semantic trajectory by enriching these regions with semantic labels.
- A spatial and temporal semantic similarity measure is given to describe the semantic similarity of two trajectories.
- A pruning strategy based on time entropy is proposed to lighten tedious calculations and comparisons.

The remaining part of this paper is structured as follows. Section 2 reviews the related work. Section 3 details the semantic trajectory enrichment approach and Section 4 presents the similarity measure of semantic trajectory. Section 5 talks about the pruning strategy based on time entropy for top-*k* search in the identification of groups of similar behaviors. Section 6 illustrates the experimental evaluations. Section 7 finally concludes this paper.

2. Related Work

In this section, we present the research of semantic trajectory, including stay point detection, trajectory semantic enrichment, semantic similarity and so on.

A semantic trajectory is a trajectory that has been enriched with annotations that provide significant semantic knowledge about movements [18-20]. The annotation contains all kinds of contextual information (such as trajectory level, attribute level and position level). For example, recording the goal of a moving object (e.g., go to school) is an annotation at trajectory level. The *bus* is a possible value for transportation means that it is an annotation at the attribute level. Moreover, recording each important position in the trajectories of a moving object is an annotation at position level (e.g., home, office). We can also attach a complex combination of various annotations to a trajectory. It offers us a better understanding of the motion of the moving object at the higher semantic representations level. What's more, the adding semantics enhances the analysis of data and facilitates the discovery of semantically implicit patterns and behaviors [21-23].

Stay point detection is a prerequisite for trajectory semantic enrichment. A simple detection method is based on the velocity. For a certain segment in a trajectory, if the velocity of the moving object is zero or very small, then it can be seen a stay segment [24]. In [25-27], a more powerful method based on the spatial threshold and temporal threshold is introduced. However, this method may result in a lot of redundant stay points in such a situation that the user walks in a small region for a long time. Another way to find stay points is to use clustering algorithms. Cao et al. used two improved OPTICS and K-means to extract stay points from the GPS records [25]. Zhou et al. developed DJ-Cluster, a density-based clustering algorithm, to discover stay places of arbitrary shape [17]. Zheng et al. and Lv et al. found the stay points clusters by applying another density-based clustering algorithm DBSCAN [28, 29]. However, most of these algorithms only handle single trajectory and cannot work well with a mass of trajectories due to high time cost.

There are various ways to assign semantic meanings to physical stay regions. Some applications require users to input semantic place labels manually [30]. Apparently, this approach relies heavily on users' initiative, and thus it does not deal well with massive stay points. Some existing works used Hidden Markov Model (HMM)-based technique to recognize the semantic annotation of stop points [19]. Another method of semantic enrichment is transforming physical locations to semantic labels based on the POIs [20, 22]. The semantics of location with these POIs (e.g., Summer Palace, Kunming Lake) is likely to be a

park. Most existing works first extract the stay points from trajectories, then select POIs labels according to these stay points, and enrich trajectories with selected semantic labels. The simple way to select the nearest POI as semantic labels, however, it is often ineffective. The intelligent optimization algorithms [31-38] may be an effective way to optimize POI selection.

Trajectory similarity refers to the similarity of two moving objects, including not only geometric patterns but also semantic abstractions extracted from the raw mobility data. By finding similar trajectories, we can identify similar users [39] or predict the next place where a user will go [24, 40]. However, most studies only focus on the spatial and temporal information and ignore the semantic information of trajectories. And up to now, few works focus on the similarity of semantic trajectories. Lv et al. proposed a framework to extract the routine activities from users daily GPS trajectories, and calculate the similarity score between users based on their routine activities [29]. Ying et al. proposed a trajectory similarity measurement named Maximal Semantic Trajectory Pattern Similarity (MSTP-Similarity), which measures the semantic similarity between trajectories [41]. They first transformed the geographic trajectories into semantic trajectories by using a geographic information database and then extracted sequential patterns from the semantic trajectories. However, it cannot calculate the similarity quickly for a massive and long-term trajectory data.

3. Semantic Trajectory Construction

3.1 Stay Regions Extraction

Stay regions extraction is to find the areas where moving objects stay for some time. These places often imply their meaningful behaviors. In this paper, we propose a fast algorithm for stay regions extraction of massive trajectories, named FSRE. It consists of two stages: extract stay points and cluster stay points.

(1) Extract stay points

The stay points can be calculated by threshold method based on space and time. For a trajectory segment(spatial threshold L), if the stay time of moving objects exceeds the temporal threshold θ in this region, then it can be seen as a stay point. The detailed process is as follows:

For a trajectory $< p_s,..., p_e >$ in time $< t_s,...,t_e >$, where p_s and p_e are starting and ending respectively, t_s and t_e are starting time and ending time respectively. Start with p_s, we first calculate the distance $l_{s,s+1}$ between p_{s+1} and p_s, then make a comparison of $l_{s,s+1}$ and L. If $l_{s,s+1} < L$, then calculate the distance $l_{s,s+2}$ between p_{s+2} and p_s until find a point p_{s+i} that meet a condition $l_{s,s+i} < L$, where $< p_s,..., p_{s+i},..., p_e >$. Calculate the temporal distance $\Delta t_{s,s+i}$ between p_{s+i} and p_s. If $\Delta t_{s,s+i} > \theta$, there is a stay point $s = (p_{mean}, t_{mean})$ in the trajectory segment $< p_s,..., p_{s+i} >$, where $p_{mean} = (lat_{mean}, lng_{mean})$, θ is the temporal threshold. The lat_{mean}, lng_{mean} and t_{mean} can be given by the following:

$$lat_{mean} = \frac{\sum_{k=s}^{s+i} lat_k}{i+1}, \quad lng_{mean} = \frac{\sum_{k=s}^{s+i} lng_k}{i+1}, \quad t_{mean} = \frac{\sum_{k=s}^{s+i} t_k}{i+1} \tag{1}$$

Instead, if $\Delta t_{s,s+i} < \theta$, there's no stay point in this segment. The p_{s+i+1} is set as a new starting. Repeat the above steps with the rest of the trajectory.

(2) Cluster stay points

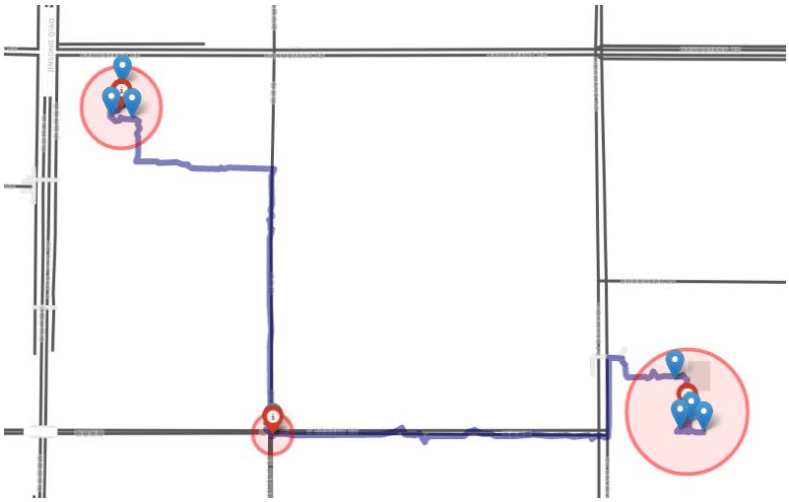


Fig. 1. Extract stay points and regions from trajectory

However, this method may result in a lot of redundant stay points. For example, moving objects move in a small region for a long time. These stay points are often concentrated in a small area. If the semantic labels of each stay points are adopted, it will generate a set of redundant labels with lots of repetitions. As shown in **Fig. 1**, there are three stay points (blue markers) at the beginning of the trajectory. The semantic labels they correspond to are the same. They are close to each other, so we can use an abstract stay point (red marker) or stay region (red region) to represent them. To find these stay regions, in this paper, we employ a variant of DBSCAN to cluster these stay points into stay region.

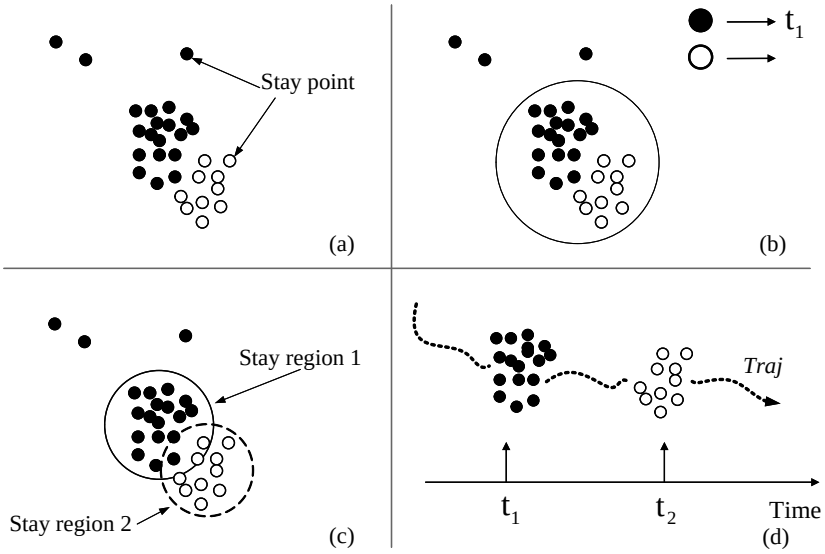


Fig. 2. The illustration of clustering stay points based on T-DBSCAN

Classical DBSCAN does not take the temporal continuity of trajectories into consideration. thus, we provided a Temporal awareness DBSCAN (denote as T-DBSCAN). **Fig. 2** is an illustration of clustering stay points based on T-DBSCAN, where black dots with timestamp t_1 and white dots with timestamp t_2 are the stay points extracted from trajectories (**Fig. 2(a)**). The black dots and white dots are nearby neighbors. But the t_1 is far away from t_2, namely,

they a long temporal distance. This indicates that the user visits the same place at different times.

In **Fig. 2(b)**, most stay points are packed into a big cluster by clustering based on spatial distance. It cannot reflect the time pattern of the trajectories. In this paper, we cluster stay points according to spatial distance and temporal distance. Only points that are close in space and time can be clustered. So these stay points need to be divided into clusters according to visited time. The *stay regions 1* (white dots cluster) and *stay regions 2* (black dots cluster) are generated by T-DBSCAN (**Fig. 2(c)**).

Finally, with the above steps, the trajectory data is transformed into an abstract $traj$: a queue of stay regions with timestamp in **Fig. 2(d)**. Each stay region contains three attributes: mean geospatial location, mean visited time and radius of region.

The POIs are semantic labels, and we can generate semantic trajectory according to these labels. Most existing works first extract the stay points from trajectories, then select POIs Labels according to these stay points, and enrich trajectories with selected semantic labels. The simple way to select the nearest POI as semantic labels. However, it is often ineffective. In this paper, for a stay region, we select all labels in it as the semantics of this region.

4. Semantic Trajectory Similarity Measure

In the previous section, the geographical trajectories are transformed into semantic trajectories. The latter consists of several collections of semantic labels with a timestamp. So each item of semantic trajectories has three attributes: spatial, temporal, and semantic information. These semantic labels represent motion of the moving object. In our study, we mainly focus on the behavior patterns. Therefore, we can get the similarity of two trajectories by calculating the similarity of collections. In this chapter, we will make a detailed description of semantic trajectory similarity.

Since stay region can be viewed as a set of semantic labels, we can use Jaccard similarity coefficient to measure the similarity of trajectories. It is defined as follows:

$$\phi(S_1, S_2) = \frac{|S_1 \bigcap S_2|}{|S_1 \bigcup S_2|} \tag{2}$$

where, the S_1 and S_2 are set of semantic labels, $S_1 =< w_{11}, w_{12}, ..., w_{1n} >$, $S_2 =< w_{21}, w_{22}, ..., w_{2n} >$, w is semantic label (e.g. school, restaurant).

For a trajectory $< p_s, ..., p_e >$, we can extract the stay points $< s_s, ..., s_e >$. Each stay point corresponds to a set of semantic labels. The starting and ending of a trajectory often have a special meaning for moving object. So these two points are seen as the stop points. Finally, the raw trajectory is transform into several collections of semantic labels with timestamp. The similarity of two trajectories can be given as follows:

$$\varphi(traj_1, traj_2) = \sum_{i=1}^{n} \phi(labels_i^1, labels_i^2) \tag{3}$$

where, the $labels_i^1$ and $labels_i^2$ are the semantic labels of i th stay point in trajectory $traj_1$ and $traj_2$. However, this approach can only measure the trajectories that have the same number of stay points. If the trajectories have a different number of stay points, for example, the $traj_1$ has N stay points, the $traj_2$ has M stay points, the similarity can't calculate directly. In order to solve this problem, we propose a merging method, For the $traj_1$ and $traj_2$, if $N > M$, $N - M$

stay points in $traj_1$ need to be merged according to the temporal distance. Firstly, the two closest points is merged, then select two closest points for merging in new set. Repeat this procedure until the number of $traj_1$ stay points is reduced to M .

In this paper, the semantic of a stay point can be represent by the collection of POIs in its region. Each POI is a triples $<type, category, name>$, such as $<restaurant, fastfood, KFC>$. This hierarchical structure is like a tree structure. For each $type$, it is a Three-tier K-ary tree, $type$ is the parent node, $category$ is the middle node and $name$ is the leaf node.

$Set_1(<restaurant, fastfood, KFC>, <restaurant, fastfood, pasta>)$

$Set_2(<restaurant, Chinesefood, noodles>, <restaurant, Chinesefood, hotpot>)$

The similarity of Set_1 and Set_2 can be given as follows:

$$\phi(Set_1, Set_2) = \frac{|Set_1 \bigcap Set_2|}{|Set_1 \bigcup Set_2|} = \frac{2}{2*1+2*2+1*4} = 0.2 \tag{4}$$

The above method can get the semantic similarity of Set_1 and Set_2, however, it doesn't take the timestamp information into consideration. And this shortcoming may have a negative on recognition accuracy. Lily goes to the supermarket at 7:30 every morning, and Bob often goes to the supermarket in the evening. At the level of semantic, they are similar. However, Lily may be a supermarket employee due to regular time. Bob may be a customer, and he just goes shopping in the evening.

To solve this problem, we propose a time coefficient to capture the temporal similarity of Set_1 and Set_2, which can be defined as follows:

$$\delta(\Delta t) = 1 - \frac{1}{\sqrt{2\pi}\sigma} + \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{\Delta t^2}{2\sigma^2}) \tag{5}$$

$$\Delta t = \begin{cases} |t_1\text{-}t_2| \bmod 12 & |t_1\text{-}t_2| < 12 \\ (24\text{-}|t_1\text{-}t_2|) \bmod 12 & \text{other} \end{cases} \tag{6}$$

where, the t_1 and t_2 are the timestamp of Set_1 and Set_2 respectively, Δt is temporal distance. $\sigma = (\Delta t^1 + \Delta t^2)/2$ is the mean stay time, Δt^1 and Δt^2 are stay time of Set_1 and Set_2 respectively in **Fig. 3**.

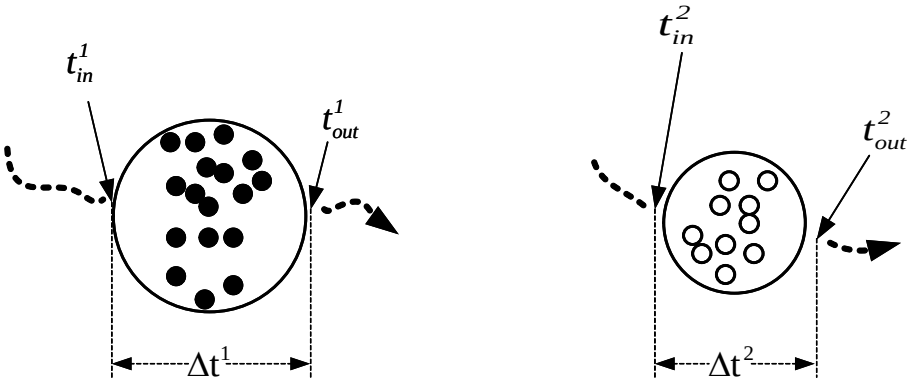


Fig. 3. The illustration of stay time

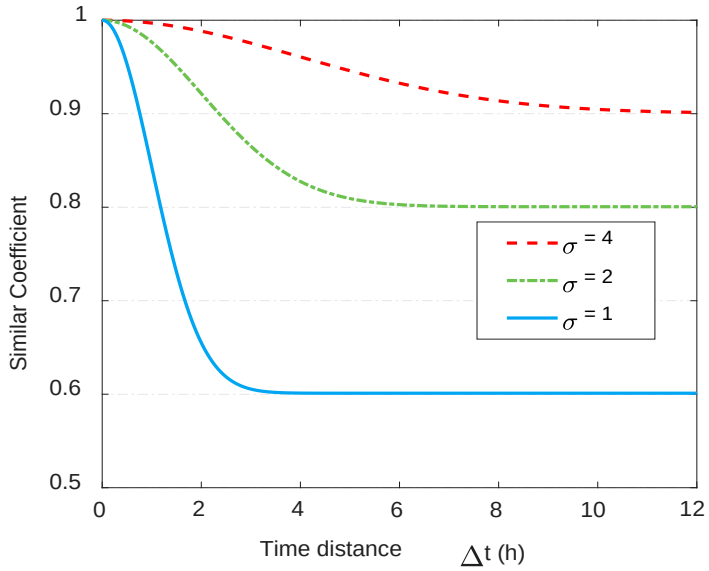


Fig. 4. Time Similar coefficient

For stay time in a region obeys the Gaussian distribution, a variant of Gaussian distribution is introduced to measure the time similarity. **Fig. 4** illustrates that the similarity coefficient as the time distance changes. The similarity coefficient decreases with time, and it has a minimum value, such as 0.6 for $\sigma = 1$, 0.8 for $\sigma = 2$. This is because if the minimum value is zero, then the similarity of trajectories is often zero. The minimum similar coefficient increases with mean stay time between Set_1 and Set_2.

Finally, the similarity of two trajectories can be given as follows:

$$\varphi(traj_1, traj_2) = \sum_{i=1}^{n} \delta_i \phi(labels_i^1, labels_i^2) \tag{7}$$

where, δ_i is time similar coefficient. In summary. for two semantic trajectories $semtraj_1$ and $semtraj_2$, the similarity can be given by **Algorithm 1**.

Algorithm 1: getSimilarity($semtraj_1$, $semtraj_2$) operation
Input: $semtraj_1$, $semtraj_2$: semantic trajectory, region sequence with semantic labels;
Output: φ: The similarity of $semtraj_1$ and $semtraj_2$;

```
Set1 ← getRegion(semtraj1);
Set2 ← getRegion(semtraj2);
N ← getNumber(Set1);
M ← getNumber(Set2);
if M > N then
    temp1=Set1; Set1=Set2; Set2=temp1;
    temp2=M; M=N; N=temp2;
while N!=M do
    Set1← mergeTimeNearestSet(Set(1));
    N=N−1;
 φ ← Equation(7);
Return(φ);
```

5. Identify Similar Behavior Groups

In this section, we proposed a pruning strategy based on time entropy to identify similar behavior groups according to massive historical trajectories. The similarity of users can be given by measuring the similarity of abstract semantic trajectories.

For two users p_1 and p_2, their similarity can be got by pairwise comparison in all semantic trajectories. However, their historical trajectories are so massive that this approach takes too much time. For example, if p_1 has m trajectories, and p_2 has n trajectories, this method needs $m*n$ comparisons. Moreover, some stay points in the trajectory are meaningless, such as traffic jam, dealing with salesman, etc. To improve efficiency of the identification, we first compress trajectories to an abstract semantic trajectory, then identify groups according to the similarity of different abstract semantic trajectories.

In reality, similar groups have similar behavior patterns, namely, similar number of stay regions. For example, the frequent trajectories of students are $\text{hom}e \rightarrow shcool$. This pattern is simple. It can be represented as two big clusters of stay points. On the contrary, taxi has no regularity in trajectories. Inspired by this, we propose a novel pruning method based on *time entropy* to detect whether a cluster of stay points is a frequent stop region. In this way, for a given user, first extract the frequent stop regions, then prune the users whose number of frequent stay regions is very different from it. Finally, the top-*k* similar users are found from candidate users.

Time Entropy, it is a measure of frequent stay regions in user's trajectories. For a stay region $Stay(s_1, s_2, ..., s_n)$ which is formed by m trajectories $Traj(traj_1, traj_2, ..., traj_m)$, the correspondence between trajectory and stay region is $< traj_i, S >$, where $traj_i \in Traj$, $S \subset Stay$ is the set of stay points which is extracted from $traj_i$. Frequent stay region is the region which contains a lot of stay points of several trajectories, and it can be measured as follows:

$$H = -\sum_{i=1}^{m} p_i \log p_i \tag{8}$$

where, $p_i = |S|/N$ is the proportion of $traj_i$ in stay region $Stay$.

A trajectory can generate multiple stay points for different stay region, namely, the stay points in a stay cluster subordinate to several trajectories. The *time entropy* can measure the quality (or confusion) of a stay region, and the higher score, the better quality. **Fig. 5** illustrates an example of time entropy.

Fig. 5. Illustration of time entropy

In **Fig. 5**, there are five stay regions which is formed by a week's trajectories (different colors). The cluster A and B have more stay points, so they can be considered as frequent stay regions. But who is better ? A contains 8 stay points, which are subordinate to seven days trajectories. B contains 10 stay points, which are subordinate to 3 days trajectories. Their *time entropy* are:

$$H_A = -\sum_{i=1}^{7} p_i \log p_i = -\frac{2}{8}\log\frac{2}{8} - 6*\frac{1}{8}\log\frac{1}{8} = 1.91 \tag{9}$$

$$H_B = -\sum_{i=1}^{3} p_i \log p_i = -\frac{3}{10}\log\frac{3}{10} - \frac{5}{10}\log\frac{5}{10} - \frac{2}{10}\log\frac{2}{10} = 1.03 \tag{10}$$

It shows A is a better frequent stay regions due to $H_a > H_b$. This is consistent with the facts that the user visits A every day.

6. Experimental Evaluation

In this section, in order to evaluate the effectiveness of the proposed method, we performed experiments on real dataset. This data set was collected in Geolife project of Microsoft Research Asia [42]. POIs come from the API interface provided of Baidu Map. All the algorithms are implemented with Scala and run on a computer with Intel Core i5-4590 CPU (3.3 GHz) and 8 GB RAM.

6.1 Experimental Setting

Dataset. The latest Geolife GPS trajectory dataset was collected in Beijing by 182 users in Beijing in a period of over five years. This dataset contains 17,621 trajectories with a total distance of 1,292,951 kilometers and a total duration of 50,176 hours. These trajectories were recorded by different GPS loggers and GPS-phones, and have a variety of sampling rates. 91.5 percent of the trajectories are logged in a dense representation, e.g. every 1～5 seconds or every 5～10 meters per point.

Parameter Setting. This experiment consists of several steps and parameters. Among them, there are spatial threshold L, temporal threshold θ. For the spatial threshold L and the temporal threshold θ, it means that the moving object stay in a certain area L for a time θ. For a given L, the smaller θ will result a lot of inappropriate stay points, conversely the longer θ will missing several important stay points. In this paper, we use the ratio of threshold $200m/10min$ in [20]. Meanwhile, the length of spatial threshold L is crucial for the performance of our proposed method as shown in **Table 1**.

Table 1. Setting of spatial threshold L and temporal threshold θ

L,θ	N_s	N_c	$\Delta t_1(ms)$	$\Delta t_2(ms)$	$\Delta d(m)$
$L=50m,\theta=2.5\min$	42	4	446	543	18.21
$L=100m,\theta=5\min$	30	4	436	446	23.03
$L=200m,\theta=10\min$	26	3	469	481	28.26
$L=500m,\theta=25\min$	24	3	504	523	32.23
$L=1000m,\theta=50\min$	21	3	573	589	46.82
Our approach	-	3	0	38902	0

In **Table 1**, we conducted an experiment on five walking trajectories of *Num*.20 at different length of L. The N_s is the number of stay points in these trajectories, it increases with the increase of L. N_c is number of stay clusters. It does not change much, which means all setting can find the right stay region. Δt_1 is the time of finding stay points, it reduces with the increase of L. This is due to the unnecessary calculation for trajectory segment. For example, there's only one small stay segment at the tail part of a $1000m$ trajectory. We just need to calculate the last segment based on threshold $L=50m$. On the contrary, for threshold $L=1000m$, the whole trajectory need to be calculated. On the one hand, these calculations are often not needed, on the other hand, it can lead to bigger distance error. The Δd is the distance error between our calculations and the center of real stay region. The Δt_2 is the time of merge stay points based on DBSCAN, it increases rapidly with the increase of the number of points.

6.2 Experimental Results

Efficiency and Robustness. Stay points search is very important to our system. A quick and accurate algorithm of finding stay regions can effectively improve system performance. In this paper, we proposed a method for quick find stay points in trajectories based on space-time threshold and the DBSCAN and made a comparative experiment on data sets of different sizes.

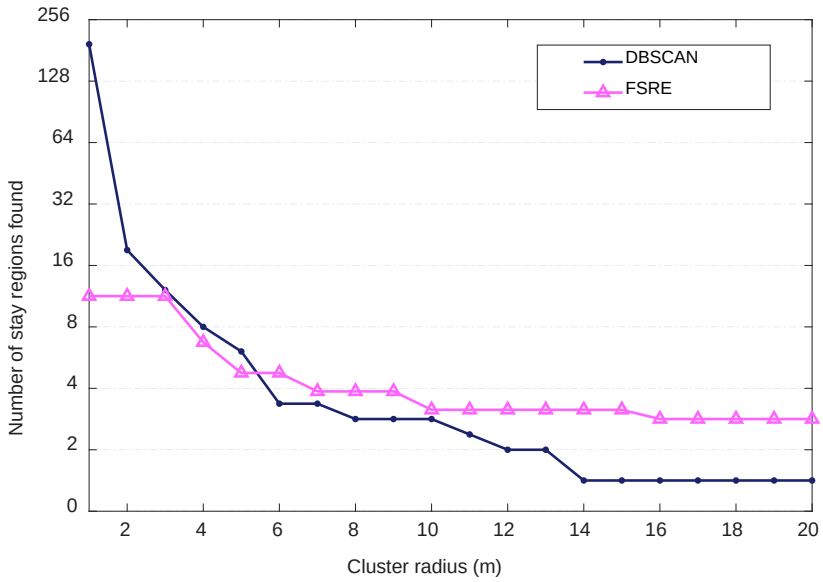


Fig. 6. Comparison of DBSCAN and FSRE on the number of stay regions found

Fig. 6 illustrates the effective comparison of the number of stay regions found as the cluster radius changes. We carried out experiments on the walking trajectory data. For the DBSCAN, it need to cluster all trajectory points, and the radius is crucial to the number of clusters found. As shown in **Fig. 6**, it has nearly 200 clusters when the radius is 1.0*m*. However, when the radius is 4.0*m*, the number of clusters reduces to 8. Finally, all trajectory points (including the points on the road) are clustered into a cluster the radius is 14.0*m*. In a walking trajectory, the distance between two points is usually 0.5-10 meters, and only a few distance will be further. So when the radius goes beyond the max distance, there is only a cluster. Even worse, the max distance decreases with the increase of the trajectory points. On the Contrary, for our approach, we just need cluster the stay points that are found by our method. Usually, these stay points have been clustered into the prototype of stay regions and the number of points is greatly reduced. In this experiment, it has only 12 clusters when the radius is 1.0*m*. As the radius increases, these stay points are clustered into three stay regions. Because of the distance between different clusters is so far, it is difficult for them to be clustered into a cluster. So our method is more robust.

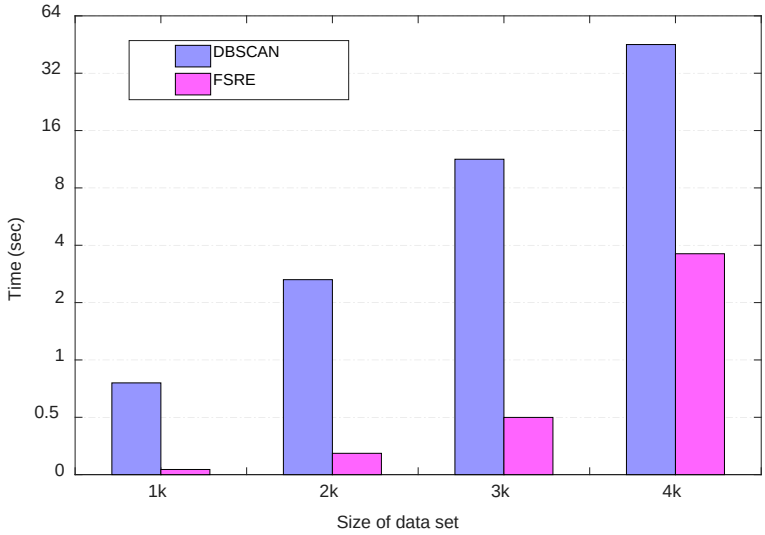


Fig. 7. Comparison of DBSCAN and FSRE on the running time

Fig. 7 illustrates the time comparison of finding stay regions. For the DBSCAN, it increases rapidly with the increase of the number of points. This is an exponential growth, so it will take a long time to find stay regions for the large data set. Obviously this is not feasible for the big data in the present system. Instead, our method can quickly find the stay regions due to that fewer stay points have been selected. On the other hand, our algorithm is more robust the DBSCAN. DBSCAN is very sensitive to the density of the trajectory points. A small change of the radius will lead to a big change of the number of clusters. Therefore, for the DBSCAN, we need to carefully adjust the radius. Instead, our method greatly reduces the number of stay points, and more importantly, these stay points have been clustered into simple clusters. The distances between different clusters are very large, so they can be clustered quickly.

Effectiveness of Semantic Trajectory. In this section, we first analyze the similarity of a person's semantic trajectories in different days of the week. The following **Table 2** shows the similarity comparison of one week trajectories of the Num.20 volunteer in Geolife project. As shown in the table, the semantic trajectories of the workday (Monday to Friday) are similar and so are the weekend's semantic trajectories. However, there are some differences between the workday and weekend. This shows that he has a regular life. On the contrary, for someone, if the trajectories are different every day, then his life pattern is not stable, or his workplace is not fixed, such as taxi drivers.

Table 2. Comparison of semantic trajectory similarity by weeks

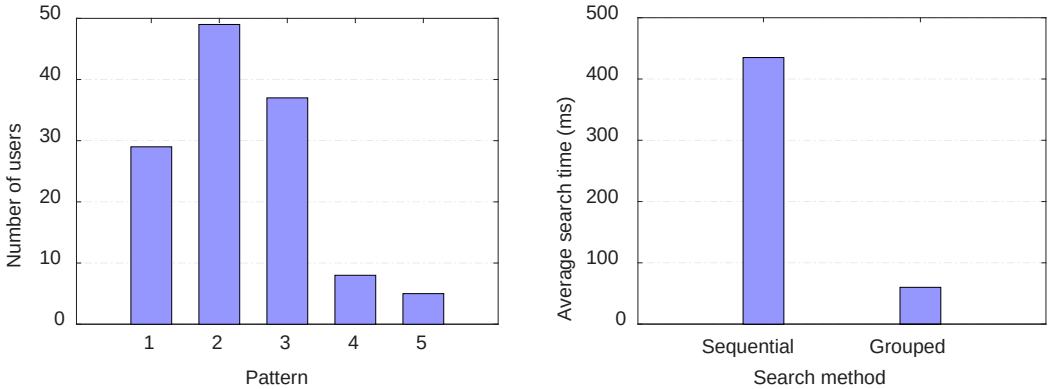
sim	MON	TUE	WED	THU	FRI	SAT	SUN
MON	1.00						
TUE	0.51/0.34	1.00					
WED	0.46/0.28	0.42/0.31	1.00				
THU	0.40/0.29	0.44/0.29	0.42/0.21	1.00			
FRI	0.49/0.31	0.48/0.43	0.36/0.38	0.46/0.26	1.00		
SAT	0.32/0.30	0.29/0.24	0.31/0.27	0.50/0.34	0.46/0.31	1.00	
SUN	0.26/0.29	0.30/0.36	0.25/0.30	0.40/0.37	0.43/0.28	0.52/0.32	1.00

Efficiency of Search. By analyzing the similarity of these semantic trajectories, we can get the similarity of two individuals. However, the users usually have so many trajectories that it's hard to get their similarity by pairwise comparison in semantic trajectories. In this paper, we introduce the time entropy to discovery the life pattern. **Fig. 8** shows the extraction process of the *Num*.20 with 151 trajectories of three months.

(a) User 20 (pattern 1) (b) User 36 (pattern 2)

Fig. 8. Illustration of different life patterns

Fig. 9 shows the distribution of life pattern in Geolife data and efficiency search. We classified the volunteers in Geolife project into five types: 0 ~ 5. We ignore the pattern for 54 volunteers whose trajectories less than a week or the pattern is zero and this process takes about an hour. Among them, 0 represents those volunteers that have no fixed stay region, and 5 represents people who have five or more frequent stay regions. For someone's top-k similar users, we first analyze his life pattern, then select the similar life pattern users as the candidates, final find the top-k semantic similar users from the candidates. Our method can greatly reduce the number of comparisons shown in **Fig. 9(b)**.



(a) Distribution (b) Query time

Fig. 9. Distribution of life pattern in Geolife data and efficiency search

7. Conclusion

In this paper, we proposed a novel trajectory semantics calculation method to identify groups of similar behaviors. A fast and efficient approach to get stay regions in daily trajectories based on time and space, then generate semantic trajectory by enriching these regions with semantic labels. Meanwhile, a spatio-temporal semantic similarity measure is given to describe the semantic similarity of two trajectories. Aiming at the problem of high complexity there are massive locus similarity, a pruning strategy based on time entropy is proposed to lighten tedious calculations and comparisons. Experimental results on real trajectory data show the effectiveness and efficiency of the proposed methods.

Acknowledgement

The work was partially supported by the National Key R&D Program of China under grant number 2017YFC0803300, National Natural Science Foundation of China under grant number 91546111 and 91646201, Beijing Municipal Education Commission Science and Technology Program under grant number KZ201610005009 and KM201610005022.

References

[1] S. Dhar, U. Varshney, "Challenges and business models for mobile location-based services and advertising," *Communications of the ACM*, vol. 54, no. 5, pp. 121-128, 2011. Article (CrossRef Link)

[2] E. Cho, S. A. Myers and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011. Article (CrossRef Link)

[3] F. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630-638, 2010. Article (CrossRef Link)

[4] N. Buch, S. A. Velastin and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920-939, 2011. Article (CrossRef Link)

[5] L. Boudjeloud-Assala, T. M. Thuy, "A clustering algorithm based on elitist evolutionary approach," *International Journal of Bio-Inspired Computation*, vol. 10, no. 4, pp. 258-266, 2017. Article (CrossRef Link)

[6] W. Pan, Y. Zhou and Z. Li, "An exponential function inflation size of multi-verse optimisation algorithm for global optimization," *International Journal of Computing Science and Mathematics*, vol. 8, no. 2, pp. 115-128, 2017. Article (CrossRef Link)

[7] X. Cai, H. Wang, Z. Cui, J. Cai, Y. Xue and L. Wang, "Bat algorithm with triangle-flipping strategy for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 2, pp. 199-215, 2018. Article (CrossRef Link)

[8] S. Zhan, Y. Zhong, Z. Zhang, D. Zhong and H. Zhang, "Comparative analysis of selection schemes used in artificial bee colony algorithm," *International Journal of Computing Science and Mathematics*, vol. 8, no. 3, pp. 218-227, 2017. Article (CrossRef Link)

[9] M. Zhang, H. Wang, Z. Cui and J. Chen, "Hybrid multi-objective cuckoo search with dynamical local search," *Memetic Computing*, vol. 10, no. 2, pp. 199-208, 2018. Article (CrossRef Link)

[10] U. Rajput, M. Kumari, "Mobile robot path planning with modified ant colony optimization," *International Journal of Bio-Inspired Computation*, vol. 9, no. 2, pp. 106-113, 2017. Article (CrossRef Link)

[11] Z. Cui, Y. Cao, X. Cai, J. Cai and J. Chen, "Optimal LEACH protocol with modified bat algorithm for big data sensing systems in Internet of Things," *Journal of Parallel and Distributed Computing*, 2018. Article (CrossRef Link)
[12] X. Chen, J. Pang and R. Xue, "Constructing and comparing user mobility profiles for location-based services," in *Proc. of the 28th Annual ACM Symposium on Applied Computing*, 2013. Article (CrossRef Link)
[13] F. Giannotti, M. Nanni, F. Pinelli and D. Pedreschi, "Trajectory pattern mining," in *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007. Article (CrossRef Link)
[14] J. H. Kang, W. Welbourne, B. Stewart and G. Borriello, "Extracting places from traces of locations," in *Proc. of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, 2004. Article (CrossRef Link)
[15] J. Niedermayer, A. Z U Fle, T. Emrich, M. Renz, N. Mamoulis, L. Chen and H. Kriegel, "Probabilistic nearest neighbor queries on uncertain moving object trajectories," *Proceedings of the VLDB Endowment*, vol. 7, no. 3, pp. 205-216, 2013. Article (CrossRef Link)
[16] L. Wei, Y. Zheng and W. Peng, "Constructing popular routes from uncertain trajectories," in *Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012. Article (CrossRef Link)
[17] C. Zhou, N. Bhatnagar, S. Shekhar and L. Terveen, "Mining personally important places from GPS tracks," in *Proc. of the 23rd International Conference on Data Engineering Workshop*, 2007. Article (CrossRef Link)
[18] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis and Others, "Semantic trajectories modeling and analysis," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, pp. 42, 2013. Article (CrossRef Link)
[19] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra and K. Aberer, "Semantic trajectories: Mobility data computation and annotation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 3, pp. 49, 2013. Article (CrossRef Link)
[20] Y. Zheng, "Trajectory data mining: an overview," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 3, pp. 29, 2015. Article (CrossRef Link)
[21] C. Guan, X. Lu, X. Li, E. Chen, W. Zhou and H. Xiong, "Discovery of college students in financial hardship," in *Proc. of the 2015 IEEE International Conference on Data Mining (ICDM)*, 2015. Article (CrossRef Link)
[22] M. Lv, L. Chen, Z. Xu, Y. Li and G. Chen, "The discovery of personally semantic places based on trajectory data mining," *Neurocomputing*, vol. 173, pp. 1142-1153, 2016. Article (CrossRef Link)
[23] L. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, W. Peng and T. L. Porta, "A framework of traveling companion discovery on trajectory data streams," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 1, pp. 3, 2013. Article (CrossRef Link)
[24] D. Ashbrook, T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous computing*, vol. 7, no. 5, pp. 275-286, 2003. Article (CrossRef Link)
[25] X. Cao, G. Cong and C. S. Jensen, "Mining significant semantic locations from GPS data," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1009-1020, 2010. Article (CrossRef Link)
[26] H. Su, K. Zheng, K. Zeng, J. Huang and X. Zhou, "STMaker: a system to make sense of trajectory data," in *Proc. of Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1701-1704, 2014. Article (CrossRef Link)
[27] Y. Zheng, L. Zhang, X. Xie and W. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. of the 18th International Conference on World Wide Web*, 2009. Article (CrossRef Link)
[28] K. Zheng, Y. Zheng, N. J. Yuan and S. Shang, "On discovery of gathering patterns from trajectories," in *Proc. of the 29th International Conference on Data Engineering (ICDE)*, 2013. Article (CrossRef Link)

[29] M. Lv, L. Chen and G. Chen, "Mining user similarity based on routine activities," *Information Sciences*, vol. 236, pp. 17-32, 2013. Article (CrossRef Link)
[30] L. Barkhuus, B. Brown, M. Bell, S. Sherwood, M. Hall and M. Chalmers, "From awareness to repartee: sharing location within social groups," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 2008. Article (CrossRef Link)
[31] V. Panthi, D. P. Mohapatra, "A framework for generating prioritised test scenarios using firefly optimisation technique," *International Journal of Computing Science and Mathematics*, vol. 8, no. 3, pp. 228-237, 2017. Article (CrossRef Link)
[32] Z. Cui, B. Sun, G. Wang, Y. Xue and J. Chen, "A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber--physical systems," *Journal of Parallel and Distributed Computing*, vol. 103, pp. 42-52, 2017. Article (CrossRef Link)
[33] X. You, Y. Ma and Z. Liu, "An improved artificial bee colony algorithm for solving parameter identification problems," *International Journal of Computing Science and Mathematics*, vol. 8, no. 6, pp. 570-579, 2017. Article (CrossRef Link)
[34] R. Sivaraj, R. D. Priya, "Bayesian-based parallel ant system for missing value estimation in large databases," *International Journal of Bio-Inspired Computation*, vol. 9, no. 2, pp. 114-120, 2017. Article (CrossRef Link)
[35] Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang and J. Chen, "Detection of Malicious Code Variants Based on Deep Learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187-3196, 2018. Article (CrossRef Link)
[36] Z. Li, G. Li, Y. Sun, G. Jiang, J. Kong and H. Liu, "Development of articulated robot trajectory planning," *International Journal of Computing Science and Mathematics*, vol. 8, no. 1, pp. 52-60, 2017. Article (CrossRef Link)
[37] L. M. Torres-Trevi N O, "Let the swarm be: an implicit elitism in swarm intelligence," *International Journal of Bio-Inspired Computation*, vol. 9, no. 2, pp. 65-76, 2017. Article (CrossRef Link)
[38] S. S. Reddy, B. K. Panigrahi, "Optimal power flow using clustered adaptive teaching learning-based optimization," *International Journal of Bio-Inspired Computation*, vol. 9, no. 4, pp. 226-234, 2017. Article (CrossRef Link)
[39] Y. Zheng, L. Zhang, Z. Ma, X. Xie and W. Ma, "Recommending friends and locations based on individual location history," *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, pp. 5, 2011. Article (CrossRef Link)
[40] J. J. Ying, W. Lee, T. Weng and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proc. of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2011. Article (CrossRef Link)
[41] J. J. Ying, E. H. Lu, W. Lee, T. Weng and V. S. Tseng, "Mining user similarity from semantic trajectories," in *Proc. of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, 2010. Article (CrossRef Link)
[42] Y. Zheng, L. Wang, R. Zhang, X. Xie and W. Ma, "GeoLife: Managing and understanding your past life over maps," in *Proc. of the 9th International Conference on Mobile Data Management*, 2008. Article (CrossRef Link)

Yang Cao is a Ph.D. candidate of College of Computer Science, Beijing University of Technology, China. He received his M.S. degree (2015) in Computer Science and Technology from Taiyuan University of Science and Technology, China. His main research interests are in the field of swarm intelligence, machine learning and big data analysis.

Zhi Cai is a Lecturer in the College of Computer Science, Beijing University of Technology, China. He obtained his Ph.D. in 2011 from the Department of Computing and Mathematics of the Manchester Metropolitan University, U.K. His research interests include Information Retrieval, Ranking in Relational Databases, Keyword Search, Intelligent Transportation Systems, and Analysis and Ontology Engineering. He has published about 20 papers in academic journals and conferences.

Fei Xue is a Lecturer at the School of Information, Beijing Wuzi University. He received the P.D. degree in Computer Science and Technology from Beijing University of Technology, China, in 2016. His current research interests are in the areas of swarm intelligence optimization, deep learning and network security. He has published about 30 papers in academic journals and conferences.

Tong Li is a Lecturer in the College of Computer Science, Beijing University of Technology, China. He obtained his Ph.D. in 2016 from the International Doctorate School in Information and Communication Technologies of the University of Trento, Italy. He has been an author or co-author of more than 30 papers in peer-reviewed journals, conferences, or workshops. His research interests are in the areas of software engineering, security engineering and conceptual modeling.

Zhiming Ding is a Professor of the College of Computer Science, Beijing University of Technology, China. He received his Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences (2002) respectively. His main research interests include database systems, spatial-temporal data management, machine learning and big data analysis. He owns 5 invention patents, and has published 3 books and about 120 papers in academic journals and conferences.