Chapter 12
Spatiotemporal Pattern Mining: Algorithms and Applications

Zhenhui Li

Abstract With the fast development of positioning technology, spatiotemporal data has become widely available nowadays. Mining patterns from spatiotemporal data has many important applications in human mobility understanding, smart transportation, urban planning and ecological studies. In this chapter, we provide an overview of spatiotemporal data mining methods. We classify the patterns into three categories: (1) individual periodic pattern; (2) pairwise movement pattern and (3) aggregative patterns over multiple trajectories. This chapter states the challenges of pattern discovery, reviews the state-of-the-art methods and also discusses the limitations of existing methods.

Keywords Spatiotemporal data · Trajectory · Moving object · Data mining

1 Introduction

With the rapid development of positioning technologies, sensor networks, and online social media, spatiotemporal data is now widely collected from smartphones carried by people, sensor tags attached to animals, GPS tracking systems on cars and airplanes, RFID tags on merchandise, and location-based services offered by social media. While such tracking systems act as real-time monitoring platforms, analyzing spatiotemporal data generated from these systems frames many research problems and high-impact applications:

- Understanding animal movement is important to addressing environmental challenges such as climate and land use change, bio-diversity loss, invasive species, and infectious diseases.
- Traffic patterns help people with fastest path finding based on dynamic traffic information; automatic and early identification of traffic incidents; and safety alerts when dangerous driving behaviors are recognized.

Z. Li (✉)
Pennsylvania State University, University Park, USA
e-mail: jessieli@ist.psu.edu

C. C. Aggarwal, J. Han (eds.), *Frequent Pattern Mining*,
DOI 10.1007/978-3-319-07821-2_12,

- Unusual vessel trajectory could be a sign of smuggling; outlying taking-off/landing patterns could be a dangerous signal for aviation; and detection of suspicious human movements could help prevent crimes and terrorism.
- Spatiotemporal interactions of human may tell the semantic relationships among them such as colleague, family or friend relationships. Different from cyber social network such as Facebook friends, spatiotemporal relationships reveal more complicated physical social network.

This book chapter discusses the state-of-art data mining methods to discover underlying patterns in movements. Various patterns, characteristics, anomalies, and actionable knowledge can be mined from massive moving object data. We will focus on following three categories of movement patterns:

- *Individual periodic pattern.* One most basic pattern in moving objects is the periodicity. Human repeat daily or weekly movement patterns. Animals have seasonal migration patterns. We will discuss how to automatically *detect the periods* in a trajectory and how to *mine frequent periodic patterns* after periods are detected. We will also describe the methods of using periodic patterns for *future movement prediction*.
- *Pairwise movement pattern.* Focusing on two moving objects only, we will discuss different trajectory *similarity measures* and the methods to mine generic, behavioral and semantic patterns. *Generic patterns* include the attraction or avoidance relationships between two moving objects. In *behavioral patterns*, we will mainly discuss how to detect the following and leadership patterns. To mine *semantic relationships*, such as colleague or friends, we will discuss the supervised learning frameworks with various spatiotemporal features.
- *Aggregate patterns over multiple trajectories.* The aggregate patterns describe a group of moving objects share similar movement patterns. *Frequent trajectory patterns* can find the frequent sequential transitions among spatial regions. *Moving object clusters*, such as flock, convoy and swarm, will detect a group of moving objects being spatially close for a relatively long period of time. *Trajectory clustering* groups similar (sub-)trajectories and reveals the popular paths shared by trajectories.

The rest of the chapter is organized as follows. Section 2 introduces the basic definitions and concepts in spatiotemporal data mining. We then study the individual periodic patterns in Sect. 3. Section 4 covers pairwise movement patterns. And we present aggregate patterns in Sect. 5. Finally, we summarize the chapter in Sect. 6.

2 Basic Concept

2.1 Spatiotemporal Data Collection

Spatiotemporal data is a broad concept. As long as the data is related to spatial and temporal information, we call it spatiotemporal data. Two most frequently seen spatiotemporal data are (1) ID-based spatiotemporal data collected from GPS and (2) location-based data collected from sensors.

Table 12.1 A sample of real moving object data showing non-constant sampling rate

Id	Timestamp	Location-long	Location-lat
2635	1997-07-24 20:50:00	– 149.007	63.809
2635	1997-07-24 21:23:35	– 148.897	63.766
2635	1997-07-27 22:30:23	– 148.967	63.824
2635	1997-07-31 02:52:48	– 149.026	63.803
2635	1997-08-03 01:47:04	– 149.046	63.795

An ID-based spatiotemporal data is essentially a trajectory. The tracking device is attached to a moving object. For example, scientists can embed sensors on animals' body and use GPS to track them; cellphone data can reveal an individual person's movement; and GPS embedded in cars can track a vehicle's movement. Suppose we have trajectories of n moving objects $\{o_1, o_2, \ldots, o_n\}$. Each trajectory is represented as a sequence of points (x_1, y_1, t_m), (x_2, y_2, t_m), $\ldots$, (x_n, y_n, t_m), where (x_i, y_i) is a location (longitude and latitude) and t_i is the time when location (x_i, y_i) is recorded. The trajectory data could contain a large set of moving objects and the tracking time for moving objects could expand several years.

A location-based spatiotemporal data is the temporal data collected from a fixed location. The tracking devices (i.e., sensors) are fixed at certain locations. For example, sensors embedded on the road can track the speed and volume of the traffic; sensors are installed at various locations to track the weather information, such as temperature, wind speed and humidity. There are a set of associated properties at location (x, y) at time t. We use $f(x, y, t, p)$ to denote the value of property p at location (x, y) at time t.

In this book chapter, we will focus on ID-based spatiotemporal data (i.e., trajectories). We will mainly discuss about the patterns of animal and human movement data.

2.2 Data Preprocessing

The raw trajectory data are unevenly sampled and could contain a long period of missing data. Table 12.1 shows a sample of raw trajectory data. As we can see that the data is sampled with uneven gaps and there could be 3–4 days missing data. Depending on different tracking scenarios, the sampling rate of movement could vary from seconds to days. For bird tracking, the data could be sampled every 3–5 days in order to save battery and make the tracking time span to several years. For vehicles, the sampling rate could be as small as seconds. For mobile phone users, there is a reported point only when the user is connecting to cellphone towers.

Most of trajectory mining methods assume the data is evenly sampled. A simple and commonly used preprocessing step is to use linear interpolation to make the data evenly gapped. If two consecutive points in a trajectory are gapped with a long time period, linear interpolation may introduce a lot of errors. For example, one data point of a human trajectory is being at home at 9 p.m. on Monday and the next point is being at home at 10 p.m. on Wednesday. If we use 1 h to linearly interpolate the missing data for these 2 days, all the points between 9 p.m. Monday to 10 p.m. Wednesday will be at home. So it is better to mark those points during the long missing period as invalid points. And when conducting pattern mining methods, we will only consider the valid points. When designing data mining methods, we should pay attention to the issue of incomplete, noisy, and unevenly sampled data. Ideally, a pattern mining method should take the raw data as input or even handle the raw data with uncertainties.

2.3 Background Information

Few moving objects move in free space. Vehicles, obviously, need to follow the road network. Planes and boats need to follow more or less the scheduled paths. Animals, which live in a more free space, are also confined to embedding landscape, such as rivers, mountains and the food resources.

When considering the background information, the mining tasks become more challenging. For example, the distance between two cars cannot be calculated simply by Euclidean distance. Similarly for animals, if there is a mountain or a big river between two animals, they could be actually far away from each other. Considering background information will result in more complex distance calculation and correspondingly require different data mining methods.

For domain experts to interpret the discovered patterns, it is important to consider the underlying geography in order to understand where, when and ultimately why the entities move the way they do. Grazing sheep, for example, may perform a certain movement pattern only when they are on a certain vegetation type. Homing pigeons may show certain flight patterns only when close to a salient landscape feature such as a rive or a highway. And, the movement patterns expressed by tracked vehicle will obviously be very dependent on the environment the vehicle is moving in, be it in a car park, in a suburb or on a highway. Thus, patterns have to be conceptualized that allow linking of the movement with the embedding environment.

3 Individual Periodic Pattern

One most common activity in moving objects is the *periodic behavior*. A periodic behavior can be loosely defined as the repeating activities at certain locations with regular time intervals. For example, bald eagles start migrating to South America in late October and go back to Alaska around mid March.

Periodic behaviors provide an insightful and concise explanation over the long moving history. For example, animal movements could be summarized using several *daily* and *yearly* periodic behaviors. Periodic behaviors are also useful for compressing movement data [3, 28, 38]. Moreover, periodic behaviors are useful in future movement prediction [17], especially for a distant querying time. At the same time,

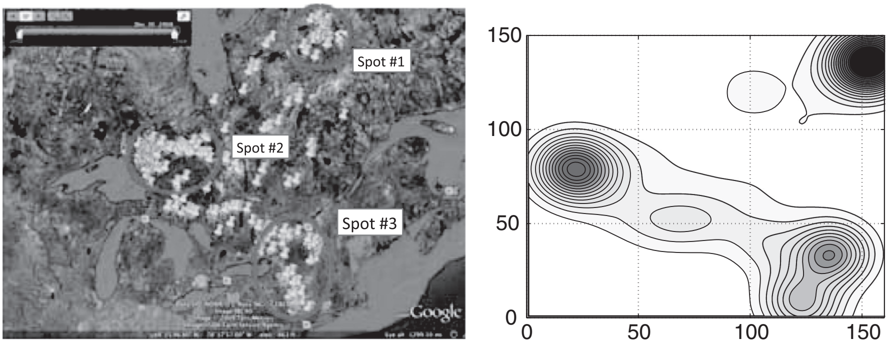

Fig. 12.1 Figure on the *left* shows the trajectory of a bald eagle over 3 years. Each *yellow pin* is a recorded GPS locations. Figure on the *right* shows the density map of all the locations in the trajectory. Periodica first detects *dense areas* as reference spots and then find periodicity for each reference spot [22]

if an object fails to follow regular periodic behaviors, it could be a signal of abnormal environment change or an accident.

In this section, we will first introduce how to automatically detect periods in a trajectory. Then, we will discuss the methods to mine frequent periodic patterns from a trajectory. Lastly, we will show how to use periodic patterns for future movement prediction.

3.1 Automatic Discovery of Periodicity in Movements

A periodic behavior can be loosely defined as the repeating activities at certain locations with *regular time intervals*. So the mining task will be, given a trajectory, find those locations and corresponding periods (i.e., regular time intervals). This is a challenging task because a real-life moving object never strictly follows a single given periodic pattern. For example, birds never follow exactly the same migration path every year. Their migration routes are strongly affected by weather conditions and thus could be substantially different from previous years. Meanwhile, even though birds generally stay in north in the summer, it is not the case that they stay at exactly the same locations on exactly the same days of the year as previous years. Therefore, "*north*" is a fairly vague geo-concept that is hard to be modeled. Moreover, birds could have multiple interleaved periodic behaviors at different spatiotemporal granularities, as a result of *daily* periodic hunting behaviors, combined with *yearly* migration behaviors.

Li et al. [22] propose Periodica to handle the aforementioned challenges. One of their key observations is that the *binary in-and-out patterns with respect to different reference spots* can reliably reveal movement periodicity. Periodica is done in two steps. In the first step, the trajectory points are clustered based on the spatial densities

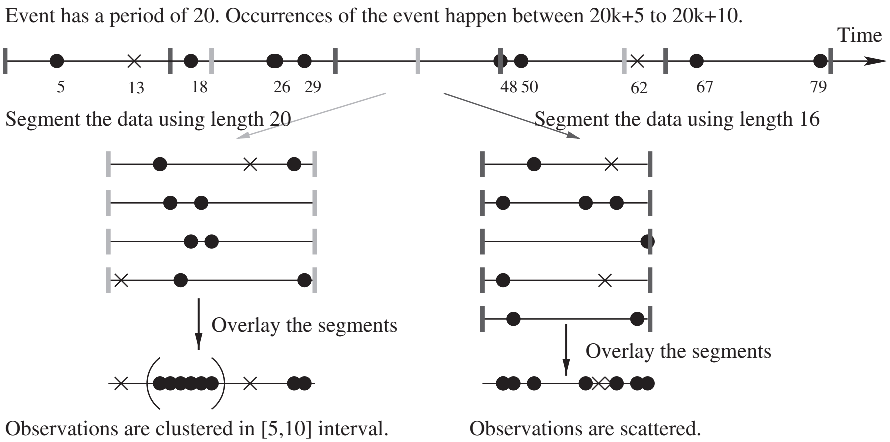

Fig. 12.2 The underlying true period is 20. The *black dots* and *cross* marks correspond to "in" and "out" events separately. When using the correct period 20 to segment and overlay the observations, as shown in the figure on the *left*, the "in" observations are clustered. Figure is from [24]

on the map to form semantic regions called *reference spots*. Figure 12.1 shows an example of an eagle movement data. There are three reference spots detected from the movement. Reference spots could be fairly large but frequently visited regions over several years, such as an area in Quebec (Spot #1 in Fig. 12.1) where birds frequently stay during the summer. In the second step, the movement is transformed into a *binary in-and-out sequence*, and then Fourier transform is applied on the sequence to detect the period.

Due to the limitations of positioning technology and data collection mechanisms, movement data collected from GPS or sensors could be *highly sparse, noisy and unsynchronized.* First, the data is often sampled at an *unsynchronized rate* (e.g., if the sampling rate of a tracking device is set to 1 h, data may be collected at 1:01, 2:08, 3:02, 4:15, and so on). Second, movement data collected can be scattered unevenly over time (e.g., collected only when the tracking device is triggered, such as the check-ins using smart phones). Third, the observations could be highly *sparse*. For example, a bird can only carry a tiny device with limited battery life. There could be only one or two reported locations in three to five days. If a sensor is not functioning or a tracking facility is turned off, it could result in a large portion of missing data. Traditional period detection methods, such as Fourier transform and auto-correlation, are known to be sensitive to such nuisances. Lomb-Scargle periodogram [27, 32] is proposed as a variation of Fourier transform to deal with unevenly spaced data, but it cannot handle the case when the data is also sparse and noisy.

Li et al. [24] develop a novel approach to detect periodicity for sparse, noisy and unsynchronized data. A "*segment-and-overlay*" idea is explored to uncover the hidden period: *Even when the observations are incomplete, the limited periodic observations will be clustered together if data is overlaid with the correct period*, as shown in Fig. 12.2. The method tries every potential periods. For a period candidate

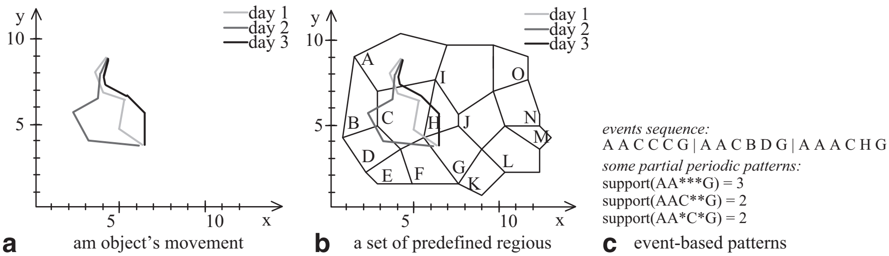

Fig. 12.3 Periodic patterns with respect to pre-defined spatial regions [28]

T, the timeline are segmented by length T and the observations are mapped to a relative timescale $[1, T]$. If T is the true period, the observations will show highly skewed distributions of the observations. Otherwise, the observations will be scattered over $[1, T]$.

3.2 Frequent Periodic Pattern Mining

Given the period, such as a day or a week, we are interested in mining the frequent regular trajectory patterns. For example, people wake up at the same time and follow more or less the same route to their work everyday. The discovery of hidden periodic patterns in spatiotemporal data, apart from unveiling important information to the data analyst, can facilitate data management substantially.

The key challenge to mine frequent pattern in movement lies in how to *transform a 2-dimensional movement sequence to 1-dimensional symbolic sequence*. As proposed by Mamoulis et al. [28], one way to handle this issue is to replace the exact locations by the regions (e.g., districts, cellphone towers, or cells of a synthetic grid) which contain them. Figure 12.3b shows an example of an area's division into such regions. By using the regions, we can transform a raw movement sequence as shown in Fig. 12.3a to an event sequence as shown in Fig. 12.3c. Now the problem becomes a traditional frequent periodic pattern mining problem [16]. In real scenario, sometimes we are interested in the *automated* discovering of descriptive regions. Mamoulis et al. [28] further propose to cluster the locations at corresponding relative timestamps, such as clustering locations at 10am over different days. They propose a top-down pattern mining method, which is more efficient than typical bottom-up method.

3.3 Using Periodic Pattern for Location Prediction

One important application of frequent periodic pattern is for future location prediction. For example, if a person repeats his periodic pattern between home and office every weekday, we could predict that this person is very likely to be in the office at

10 a.m. and to be at home at 10 p.m. Most existing techniques target at *near* future movement prediction, such as next minute or next hour. Linear motion functions [30, 31, 34, 35] have been extensively studied for movement prediction. More complicated models are studied in [36]. As pointed out by Jeung et al. [17], the actual movement of a moving object may not necessarily comply with some mathematical models. It could be more complicated than what the mathematical formulas can represent. Moreover, such models built based on recent movement are not useful for predicting distant future movement, such as next day or one month after.

Periodic patterns can help better predict future movement, especially for a *distant* query time. In [17], a prediction method based on periodic pattern is proposed. The prediction problem assumes that the period T and periodic patterns are already given. To answer predictive queries efficiently, a trajectory pattern tree is proposed to index the periodic patterns. In [17], they use *a hybrid prediction algorithm* that provides predictions for both near and distant time queries. For non-distant time queries, they use the forward query processing that treats recent movements of an object as an important parameter to predict near future locations. A set of qualified candidates will be retrieved and ranked by their premise similarities to the given query. Then they select top-k patterns and return the centers of their consequences as answers. For a distant time queries, since recent movements become less important for prediction, the backward query processing is used. Its main idea is to assign lower weights to premise similarity measure and higher weights to consequences that are closer to the query time in the ranking process of the pattern selection.

4 Pairwise Movement Patterns

In this section, we focus on pattern mining methods on two moving objects. The pairwise movement patterns are between two moving objects R and S. The trajectories of two moving objects are denoted as $R = r_1 r_2 \ldots r_n$ and $S = s_1 s_2 \ldots s_m$, where r_i and s_i are the locations of R and S at the ith timestamp.

We first introduce different similarity measures between two trajectories. Then, based on properties of patterns, we will introduce generic patterns, behavioral patterns, and semantic patterns. Generic patterns describe the overall attraction and avoidance relationship between two moving objects. Behavioral patterns describe a specific type of relationships in a (short) period of time, such as leading and following. Semantic patterns tell the semantics of a relationship (e.g., colleague and friend) in a supervised learning framework.

4.1 Similarity Measure

One way to infer the relationship strength of two moving objects is to measure the similarity of their trajectories. The simplest way of measuring the similarity between two trajectories is to use p-norm distance. The p-norm distance between trajectories of R and S is defined as:

$$L_p(R, S) = \left(\sum_{i=1}^{n} (r_i - s_i)^p \right)^{\frac{1}{p}}.$$

The p-norm distance requires the trajectory length to be the same, i.e., $n = m$. The well-known Euclidean distance and Manhattan distance are p-norm distance when $p = 2$ and $p = 1$ respectively.

The p-norm distance is easy to compute, but is sensitive to the time shift. Dynamic Time Warping (DTW) [39] can handle the local time shifting and it does not need the trajectories to be the same length. DTW is defined as:

$$DTW(R, S) = dist(r_1, s_1) + \min \left(\begin{array}{l} DTW(R[2:n], S[2:m]), \\ DTW(R[2:n], S), \\ DTW(R, S[2:m]) \end{array} \right).$$

Edit distance with Real Penalty (ERP) [4] introduces a constant value g as the gap of edit distance and uses real distance between elements as the penalty to handle local time shifting. ERP is defined as:

$$ERP(R, S) = \min \left(\begin{array}{l} ERP(R[2:n], S[2:m]) + dist(r_1, s_1), \\ ERP(R[2:n], S) + dist(r_1, g), \\ ERP(R, S[2:m]) + dist(s_1, g) \end{array} \right).$$

The Longest Common Subsequences (LCSS) [37] requires a threshold ϵ to be established. The threshold is used to determine whether or not two elements match and it allows LCSS to handle noise by quantizing the distance between two elements to two values, 0 and 1, to remove the larger distance effects caused by noise. LCSS is defined as:

$$LCSS(R, S) = \begin{cases} LCSS(R[2,n], S[2,m]) + dist(r_1, s_1) & dist(r_1, s_1) \le \epsilon \\ \max\{LCSS(R[2,n], S), LCSS(R, S[2,m])\} & otherwise \end{cases}$$

Edit Distance on Real sequence (EDR) is defined similar to LCSS except EDR assigns penalties to the gaps between two matched sub-trajectories according to the lengths of gaps. EDR is defined as:

$$EDR(R, S) = \min \left(\begin{array}{l} EDR(R[2:n], S[2:m]) + subcost, \\ EDR(R[2:n], S) + 1, \\ EDR(R, S[2:m] + 1 \end{array} \right),$$

where $subcost = 0$ if $dist(r_1, s_1) \le \epsilon$ and $subcost = 1$ otherwise.

A comparison of the similarity measures is shown in Table 12.2. All the measures except Euclidean distance can handle local time shifting. And only Euclidean distance requires the lengths of two trajectories to be the same. LCSS and EDR are more robust

Table 12.2 Summary of similarity measures [5]

Distance	Local time shifting	Noise	Metric	Computation cost
Euclidean			$\surd$	$O(n)$
DTW	$\surd$			$O(n^2)$
ERP	$\surd$		$\surd$	$O(n^2)$
LCSS	$\surd$	$\surd$		$O(n^2)$
EDR	$\surd$	$\surd$		$O(n^2)$

to noises because it does not require every point in R to be matched with a point with S. If r_i is a noise point, LCSS and EDR will skip it and assign a mismatch penalty to it. Euclidean distance and ERP are metric distances since they obey triangle inequality. Thus, efficient indexing and retrieval can be achieved by using these two distance measures.

The distance measures mentioned above are suitable to find similar trajectory with similar shapes. They can be applied on trajectories, such as hurricane trajectories and animal migration paths. To measure the similarity on human movements, it could make more sense to look at the co-locating frequency instead of trajectory shape. The meeting frequency [25] is defined as the number of timestamps that their locations are with distance ϵ:

$$freq(R, S) = \sum_{i=1}^{n} \tau(r_i, s_i),$$

where $\tau(r_i, s_i) = 1$ if $dist(r_i, s_i) \leq \epsilon$ and $\tau(r_i, s_i) = 0$ otherwise.

The similarity between two moving objects can also be measured by transitions patterns. Li et al. [20] propose to measure the similarity of two mobile users based on their location histories. The trajectory is first symbolized using the interesting locations mined from user trajectory. Given two symbolized sequences $seq_1 = r_1(k_1) \xrightarrow{\Delta t_1} r_2(k_2) \xrightarrow{\Delta t_2} \ldots r_m(k_m)$ and $seq_2 = s_1(k_1') \xrightarrow{\Delta t_1'} s_2(k_2') \xrightarrow{\Delta t_2'} \ldots s_m(k_m')$, where Δt denotes the transition time between locations and k is the number of times that the user stays in a location, seq_1 and seq_2 are similar if the following constraints are satisfied:

1. $\forall 1 \leq i \leq m, r_i = s_i$;
2. $\forall 1 \leq i \leq m, \mid \Delta t_i - \Delta t_i' \mid \leq t_{th}$, where t_{th} is a time threshold on the transition times.

4.2 Generic Pattern

Relationships between two moving objects can be classified as attraction, avoidance or neutral. In an *attraction* relationship, the presence of one individual causes the other to *approach* (i.e., reduce the distance between them). As a result, the individuals have a higher probability to be spatially close than expected based on chance. On the other hand, in an *avoidance* relationship, the presence of one individual causes the other to *move away*. So the individuals have a lower probability to be spatially close than expected. Finally, with a *neutral* relationship, individuals *do not alter* their movement patterns based on the presence (or the absence) of the other individual. So the probability that they are being spatially close is what would be expected based on independent movements.

The attraction relationship is commonly seen, for example, in animal herds or human groups (e.g., colleague and family). In addition, the avoidance relationship also naturally exists among moving objects. In animal movements, prey try to avoid predators, and different animal groups of the same species tend to avoid each other. Even in the same group, subordinate animals often avoid their more dominant group-mates. In human movements, criminals in the city try to avoid the police, whereas drug traffickers traveling on the sea try to avoid the patrol.

Intuitively, similar trajectories could be an indication of attraction relationship. The similarity can be defined by the similarity measures mentioned in the previous subsection. The assumption here is that the smaller the distance is or the higher the meeting frequency is, the stronger the attraction relationship is. Unfortunately, such assumption is often violated in real movement data. For example, two animals may be observed to be spatially close for 10 out of 100 timestamps. But is this significant enough to determine the attraction relationship? Further, another two animals are within spatial proximity for 20 out of 100 timestamps. Does this mean that the latter pair has a more significant attraction relationship than the former pair? Finally, if two animals are never being spatially close, do they necessarily have an avoidance relationship?

Li et al. [25] propose to mine significant attraction and avoidance relationships by looking into the *background territories*. The relationships are detected through the *comparison* between how frequent two objects are *expected* to meet and the *actual* meeting frequency they have. Intuitively, if the actual meeting frequency is smaller (or larger) than the expectation, the relationship is likely to be avoidance (or attraction).

Given two trajectories R and S, the probability for one point r_i in R to be spatially close to any point in S is $\frac{1}{n}\sum_{j=1}^{n}\tau(r_i, s_j)$. Then the expected meeting frequency between randomly shuffled R and S is:

$$E[freq(\sigma(R), \sigma(S))] = \sum_{i=1}^{n}\left(\frac{1}{n}\sum_{j=1}^{n}\tau(r_i, s_j)\right) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\tau(r_i, s_j),$$

where $\sigma(\cdot)$ denotes a random shuffled trajectory.

However, by comparing the actual meeting frequency with the expected meeting frequency, one cannot determine a universal *degree* of the relationship. To further measure the degree, let $\mathcal{F} = \{freq(R, \sigma(S)) \mid \sigma\}$ be the multiset of all randomized meeting frequencies. The significance value of attraction (or avoidance) between to

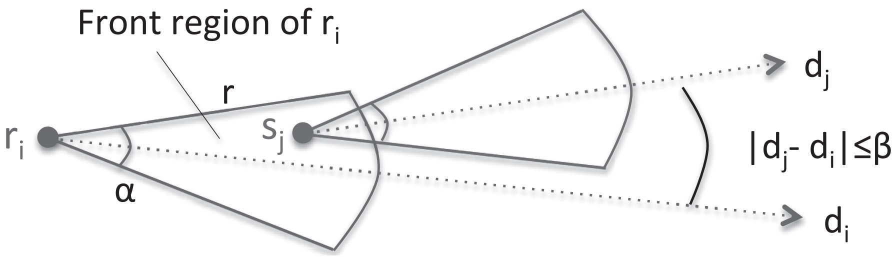

Fig. 12.4 Front region defined in [2]

moving objects R and S is defined as:

$$\begin{aligned} sig_{attract} &= Pr[freq(R,S) > freq(R,\sigma(S))] + \\ &\quad \tfrac{1}{2} Pr[freq(R,S) = freq(R,\sigma(S))], \\ sig_{avoid} &= Pr[freq(R,S) < freq(R,\sigma(S))] + \\ &\quad \tfrac{1}{2} Pr[freq(R,S) = freq(R,\sigma(S))]. \end{aligned}$$

Permutation test is conduced to get the multiset $\mathcal{F}$. Permutation test is a popular non-parametric approach, to performing hypothesis tests and constructing confidence intervals. The null hypothesis is that the movement sequences of two objects are independent. Since the total number of permutations is factorial, Monte Carlo sampling is used to approximate the significance value.

4.3 Behavioral Pattern

The behavioral patterns describe certain behaviors within a (short) period of time, such as pursuit, evasion, fighting, and play [7]. Following/leading is one interesting behavioral pattern between two moving objects. For example, animal scientists study which individual animal leads the group when animals move in order to determine the social hierarchy, whereas police and security officers look suspicious movements of a criminal who is following a victim.

Intuitively, a follower has similar trajectories as its leader but always arrives at a location with some time lag. The challenges lay in three aspects: (1) the following time lag is usually unknown and varying; (2) The follower may not have exactly the same trajectory as the leader; and (3) the following relationship could be subtle and always happens in a short period of time.

Andersson et al. [2] propose the concept of *front region*. A point s_i in the front region of r_i is defined by an apex angle α, a radius r, and an angle β restricting their difference in direction $\| d_i - d_j \|$. Figure 12.4 shows an illustration of the front region. In [2], a leader should appear in the front region of the follower(s) for at least k consecutive timestamps.

In real scenario, a leader does not necessarily appear in the front region of the followers for *consecutive* timestamps. Figure 12.5 illustrates a counter example. In

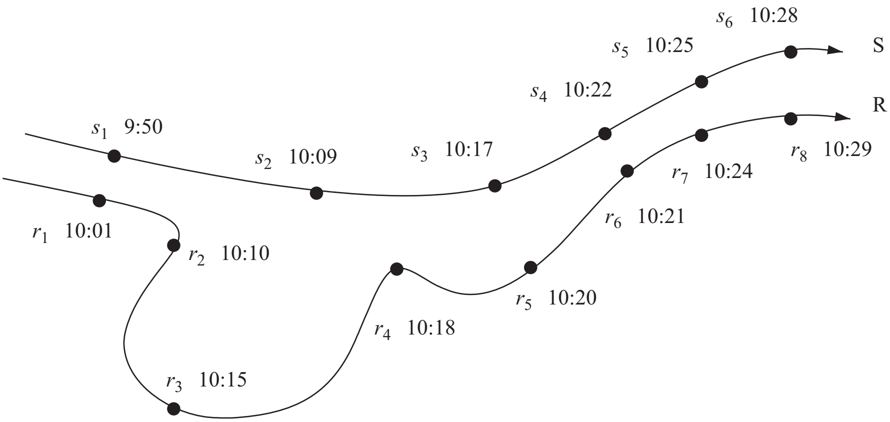

Fig. 12.5 In this example from [26], object R follows object S from *10:01* to *10:20* and moves together afterwards. Even though R follows S, s_2 is not in the front region of r_2

this example, r_2 is heading downwards at 10:10, and s_2 is apparently not in the front region of r_2. The definition of front region and the constraint on being in the front region for k consecutive timestamps are too strict to make the method applicable on real data.

Li et al. [26] propose a more relaxed definition of following pattern. Given thresholds d_{max} and l_{max}, a location pair (r_i, s_j) is said to be a *following pair* if $\| r_i - s_j \| < d_{max}$ and $0 < i - j \leq l_{max}$. By considering a following pair as a matching, the problem can be mapped to local sequence alignment (LSA) problem. Smith-Waterman algorithm [33] can be applied to find the longest following interval (best local alignment).

However, experimental results show LSA is sensitive to the parameter d_{max}. To address the problem, Li et al. [26] further propose the concept of local distance minimizer. The intuition is that if object R is following S at timestamp i, then there must exit a *strictly positive integer* $\Delta(i)$ such that r_i is spatially close to $s_{i-\Delta(i)}$. In fact, the distance between r_i and S should be minimized *locally* at such $\Delta(i)$. Based on the intuition, $f(i)$ is defined as whether r_i is following s_i at timestamp i:

$$f(i) = \begin{cases} 1, & \text{if}\, \Delta(i) > 0 \,\text{and}\, \|r_i - s_{i-\Delta(i)}\| < d_{\max} \\ 0, & \text{if}\, \Delta(i) \leq 0 \,\text{and}\, \|r_i - s_{i-\Delta(i)}\| < d_{\max} \\ \times, & \text{if}\, \|r_i - s_{i-\Delta(i)}\| \geq d_{\max} \end{cases}$$

Then the following interval $[s, t]$ should make $\sum_s^t f(i)$ maximized. The problem can be transformed to the well-known Maximum Sum Segment problem and all the following intervals can be found in linear time.

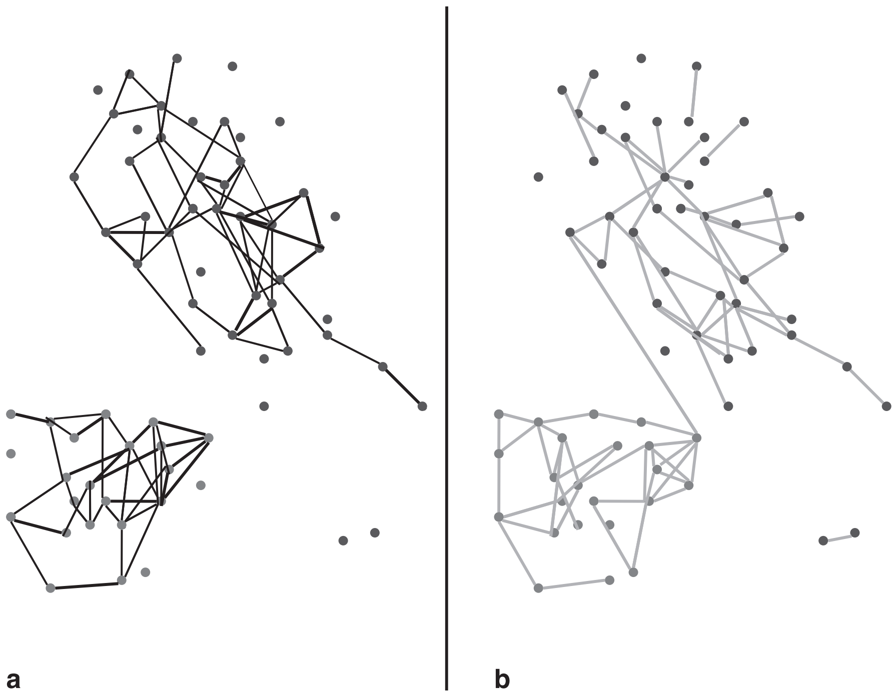

Fig. 12.6 Social networks in reality mining dataset [10]. **a** shows the inferred weighted friendship network, where the weights correspond to the relationship strengths. **b** shows the reported friendship network

4.4 Semantic Patterns

The social structure of human is one of the fundamental questions in social science. As traditional survey methods often suffer from its limited scale, Eagle and Pentland [8, 10] propose a mobile sensing framework to use human mobility data as indicators of human social network. The Reality Mining project `http://reality.media.mit.edu/` tracked the movement of 94 users for one academic year and conducted survey about the relationships between those users. The studies show that human mobility patterns strongly correlate with relationships among people.

Figure 12.6 shows two social networks of all the participants in the study. Network in Fig. 12.6a is constructed from mobility data and network in Fig. 12.6b is constructed using survey data. These two networks inferred from different data show similar structure. Such observation provides strong evidence that human movement data reflects social relationship done by survey. Later in [9], Eagle and Pentland further propose to use Principle Component Analysis (PCA) to extract representative behaviors of an individual and of groups.

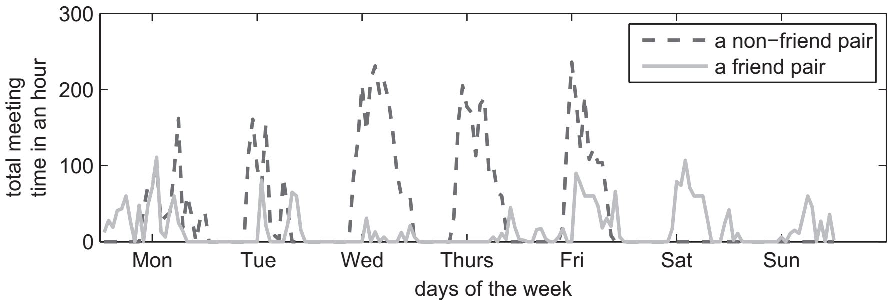

Fig. 12.7 Meeting frequency for a friend and a non-friend pair in Reality Mining dataset [23]

Category	Variables	Description	Co-location	User mobility
Intensity and Duration	NumObservations	The total number of observations of the user.		√
	NumColoc, NumColocEvening, NumColocWeekend	The number of co-location observations of the two users, in total, in the evening only, and on weekends only.	√	
	NumLocations, NumLocationsEvening, NumLocationsWeekend	The number of distinct grid boxes where the user or users were observed, in total, in the evening only, and on weekends only.	√	√
	NumHours, NumWeekdays, NumDates	The number of distinct hours of the day, days of the week, and calendar dates that the two users were observed together.	√	
	ObservationTimeSpan	The difference in seconds between the last and the first location or co-location observation.	√	√
	BoundingBoxArea	The area of the minimal axis aligned rectangle that contains the locations/co-location observations of the user/users.	√	√
Location Diversity	AvgEntropy, MedEntropy, VarEntropy, MinEntropy, MaxEntropy	The mean/median/variance/min/max of the location entropy at each location/co-location observation of the user/users.	√	√
	AvgFreq, MedFreq, VarFreq, MinFreq, MaxFreq	The mean/median/variance/min/max of the location frequency at each location/co-location observation of the user/users.	√	√
	AvgUserCount, MedUserCount, VarUserCount, MinUserCount, MaxUserCount	The mean/median/variance/min/max of the location user count at each location/co-location observation of the user/users.	√	√
Mobility Regularity	SchEntropyL, SchEntropyLH, SchEntropyLD, SchEntropyLHD	The schedule entropy of the user with respect to location, location and hour, location and day of the week, and location and hour and day of the week.		√
	SchSizeLH, SchSizeLD, SchSizeLHD	The schedule size of the user with respect to location and hour, location and day of the week, and location and hour and day of the week.		√
Specificity	AvgTFIDF, MinTFIDF, MaxTFIDF	The mean/minimum/maximum of the location TFIDF at each co-location of the two users.	√	
	PercentObservationsTogether	The total number of co-locations of the two users divided by the sum of each users total number of observations.	√	
Structural Properties	NumMutualNeighbors	The number of people who have been co-located with both users.	√	
	NeighborhoodOverlap	The number of people who have been co-located with both users divided by the number of people who have been co-located with either user.	√	
	LocationOverlap	The total number of distinct places visited by both users divided by the total number of places visited by either users.	√	

Fig. 12.8 Names and descriptions of the mobility features used in [6]

The co-locating times could be a discriminative feature to indicate the semantic relationships. Figure 12.7 shows the meeting frequency with respect to different days of the week for a friend pair and for a non-friend pair in Reality Mining dataset. It is shown in the figure that the friend pair meets more on the weekends, while the non-friend pair meets more during the weekdays. Motivated by this observation, Li et al. [23] propose to mine discriminative time intervals to classify whether two people are friends. The discriminative interval, namely T-Motif, is the time interval where there is a significant difference in meeting frequency between friend pairs and non-friend pairs.

To study how interactions in mobility data correlate with friendships on social networks, Cranshaw et al. [6] propose to build a supervised learning framework using features extracted from mobility data to predict the online relationship. They use a location sharing application based on user check-ins on Facebook to obtain the mobility data from 489 users. Using the mobility data, they propose a set of features as shown in Fig. 12.8. The features can be divided into four categories:

- **Intensity and Duration:** These features quantify the duration and the number of times that users engage in the system. This set of features includes number of observations, number of co-location observations, time spent at each location.
- **Location Diversity:** These features aim to understand the context of all locations. The features include location frequency and the location entropy. For a location L, the location entropy is defined as $Entropy(L) = -\sum_{u \in U} P_L(u) \log P_L(u)$, where U is the set of all users, and $P_L(u)$ is the probability for a user u being at the location L.
- **Specificity:** These features measure whether two persons meet at locations where less frequently visited by the public. The tf-idf score penalizes the popular places that many people frequently visit.
- **Structural Properties:** These features aim to capture network property of two users such as mutual neighbors and location overlaps.

The experimental results in [6] shows that using a variety of classification methods such as random forests and support vector machines can achieve precision above 60 % in predicting the online relationships using the mobility features.

5 Aggregate Patterns over Multiple Trajectories

The aggregate patterns describe common paths shared by a set of trajectories or a cluster of moving objects being spatially close for a long time. In this section, we first introduce the *trajectories patterns*, which is a concise description of frequent behaviors in terms of space and time. Then we will present the methods on mining moving object clusters. Finally, we discuss trajectory clustering methods.

5.1 *Frequent Trajectory Pattern Mining*

A frequent trajectory pattern is a popular path repeated by many trajectories. Finding frequent trajectory patterns is helpful in summarizing the historical trajectories and predicting the future movements. A trajectory pattern [14] is used to describe a set of individual trajectories visiting the same sequence of places with similar travel times. In trajectory patterns, two notions are important: (1) the geographical locations and (2) the travel time between locations.

If we assume the locations are already symbolized, frequent sequential pattern [1] can be considered as a simplified trajectory pattern. For example, if many people go from location X, to Y and then to Z, $X \rightarrow Y \rightarrow Z$ will a frequent sequential pattern. In order to enrich the sequential patterns with *transition time information* between locations, Giannotti et al. [13] propose the temporally annotated sequences (TAS). TAS has the following form:

$$T = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} s_n,$$

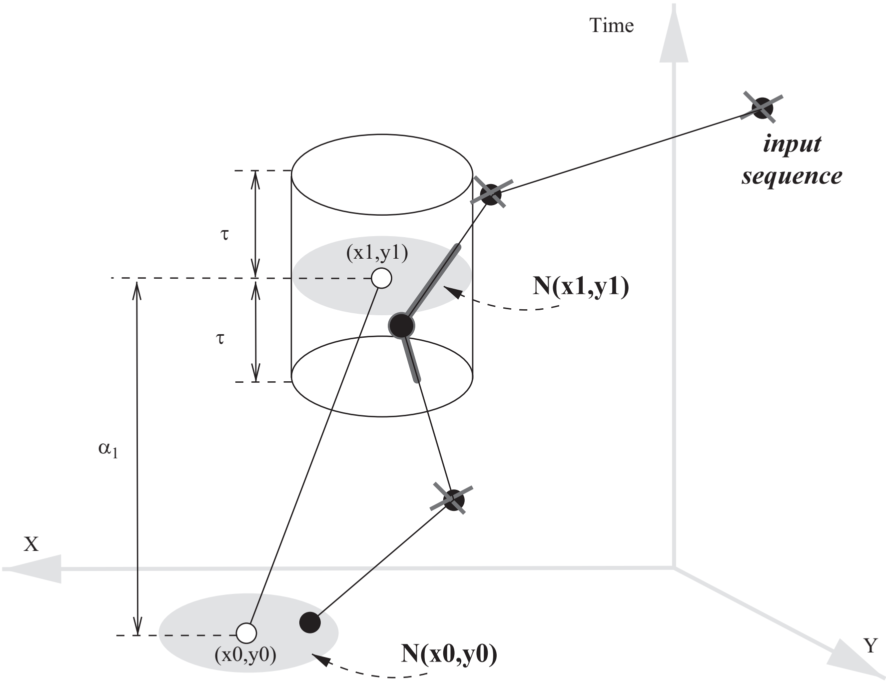

Fig. 12.9 Spatiotemporal containment of input sequence on trajectory pattern $(x_0, y_0) \xrightarrow{\alpha_1} (x_1, y_1)$ [14]

where $S = \langle s_0, \cdots, s_n\rangle$ are the elements in the sequence and $A = \langle \alpha_1, \cdots, \alpha_n\rangle$ are annotated transition time. With TAS, the pattern could be in the format of $X \xrightarrow{30\ min} Y \xrightarrow{20\ min} Z$.

Trajectory pattern [14] is defined in the same fashion of TAS where each element in S should be a spatial location:

Definition 12.1 (T-pattern) *A* Trajectory pattern*, called T-pattern, is a pair* (S, A)*, where* $S = \langle (x_0, y_0), \cdots, (x_k, y_k)\rangle$ *is a sequence of points in* $\mathbf{R}^2$*, and* $A = \langle \alpha_1, \cdots, \alpha_k\rangle \in \mathbf{R}_+^k$ *is the temporal annotation of the sequence.*

To judge whether a trajectory *contains* a trajectory pattern, Giannotti et al. [14] propose a definition on spatiotemporal containment. In Fig. 12.9, input trajectory sequence $S_1 \ldots S_5$ contains trajectory pattern $(x_0, y_0) \xrightarrow{\alpha_1} (x_1, y_1)$, because for each point (x_i, y_i) in trajectory pattern, there is a point in trajectory S that is close to it. For example, point S_3 is close to point (x_1, y_1) because it is in the spatial neighborhood (i.e., $N(x_1, y_1)$) and also the time difference between (x_1, y_1) and S_3 is less than threshold τ. Many approaches can be used as a neighborhood function $N(\cdot)$. One possible neighborhood function is to use the Regions-of-Interest (RoI) to naturally partition the space into meaning areas. If prior knowledge is not available, RoI can also be defined as the frequently visited locations/regions mined from the trajectories.

The trajectory pattern mining problem consists of finding all frequent T-patterns, such that

$$support(S, A) \geq sup_{min},$$

where $support(S, A)$ is the number of input trajectories containing the T-pattern $T(S, A)$ and the sup_{min} is a minimum support threshold.

To mine frequent T-patterns, the method to mine temporally annotated sequences (TAS) [13] can be applied if we first symbolize the locations using RoI. In [14], Giannotti et al. further discuss how to dynamically identify the locations and transition time in the pattern.

In [29], Monreale et al. propose WhereNext, a location prediction method using T-Patterns. A decision tree, named T-pattern Tree, is built and evaluated in a supervised learning framework. The tree is learned from the T-Patterns and it is used as a predictor of the next location by finding the best matching path in the tree. Different from [17] using individual frequent periodic pattern, as we discussed in Sect. 3, WhereNext [29] uses the overall traffic flows to predict the next location.

5.2 Detection of Moving Object Cluster

Moving object clusters detect groups of moving objects being spatially close for a considerably long time. Clusters of moving objects can reveal underlying communities, such as the social groups of animals or humans, and can also indirectly identify outliers that do not conform to general group behaviors.

In this section, we will discuss patterns *flock* [15], *convoy* [18] and *swarm* [21]. A moving object cluster can be loosely defined as a set of moving objects being spatially close for k timestamps. The differences among flock, convoy and swarm lie in the definitions of "spatially close" and "k (non-)consecutive timestamps".

Gudmundsson et al. [15] first propose the concept of flock.

Definition 12.2 (Flock) *A set of moving objects O form a flock for timestamps T if (1) for every timestamp in T, there is a disc with radius r containing all the objects in O; and (2) T is consisted of at least k consecutive timestamps.*

In Fig. 12.10, o_3 and o_4 form a flock since they are in the same disc from t_1 to t_4. Since flock defines spatial constraint as a fixed-radius disc, such definition might be too strict and is independent of data distribution. For example, at timestamp t_1 in Fig. 12.10, all the objects are in a density-connected cluster but using a disc may split them into multiple clusters. To relax the rigid restriction on the disc-shape cluster, Jeung et al. [18] proposes a new concept *convoy* to discover arbitrary-shape clusters. Convoy uses DBSCAN [11] to cluster points in each timestamp. Two objects in a cluster are density-connected to each other, if only there exists a sequence of objects that connect them together. The definition of density-connected permits us to capture a group of connected points with arbitrary shape.

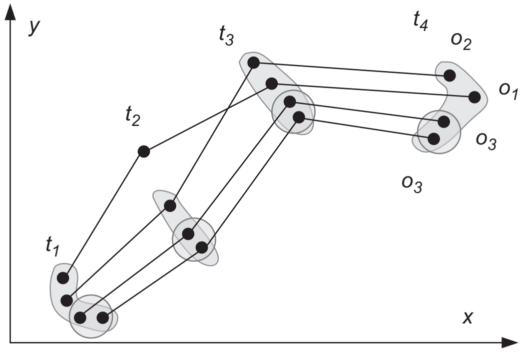

Fig. 12.10 An example of flock [15], convoy [18] and swarm [21]. If we set time constraint $k = 3$ (i.e., number of timestamps being spatially close), o_3 and o_4 form a flock since they are in a disc for four consecutive timestamps. o_1, o_3 and o_4 form a convoy since they are in the same density-connected cluster for four consecutive timestamps. Swarm considers all these four objects as a cluster since it treats o_1 at t_2 as a short deviation from the cluster

Table 12.3 Summary of moving object clusters

Pattern	Spatial constraint	Temporal constraint
Flock [15]	Disc shape	k consecutive timestamps
Convoy [18]	Arbitrary shape	k consecutive timestamps
Swarm [21]	Arbitrary shape	k (non-)consecutive timestamps

Definition 12.3 (Convoy) *A set of moving objects O form a convoy for timestamps T if (1) for every timestamp in T, all the objects in O are in the same density-connected cluster; and (2) T is consisted of at least k consecutive timestamps.*

In Fig. 12.10, three objects o_1, o_3 and o_4 are in the same density-connected cluster during the time interval $[t_1, t_4]$. Although the convoy model is much flexible than the flock, the time constraint on *k consecutive* timestamps is still too strict. The moving objects may temporarily leave the group. For example, o_1 temporarily leaves the group at t_2. If we enforce the "consecutive" time constraint, o_1 is not considered to be in the same group with other objects. Motivated by this important observation, Li et al. [21] propose the concept of *swarm* to relax the time constraint. Instead of requiring the objects being in the same cluster for *consecutive* timestamps, swarm allows the timestamps to be *non-consecutive*.

Definition 12.4 (Swarm) *A set of moving objects O form a swarm for timestamps T if (1) for every timestamp in T, all the objects in O are in the same density-connected cluster; and (2) T is consisted of at least k timestamps that are not necessarily consecutive.*

In Fig. 12.10, we can see that all the objects form a group even though o_1 temporarily leaves the cluster at t_2. Swarm is able to capture $\{o_1, o_2, o_3, o_4\}$ as one cluster.

Table 12.3 summarizes the three different patterns: flock, convoy and swarm. The definition of the swarm is the most flexible one in terms of the spatial and

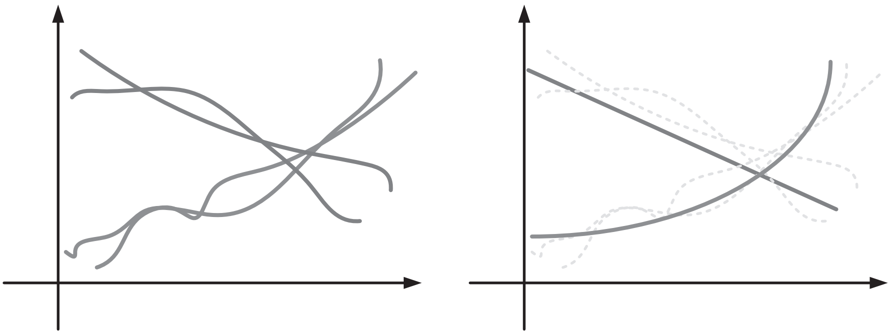

Fig. 12.11 Trajectory clustering example. Four trajectories are clustered into two clusters based on trajectory similarity

temporal constraint. The time complexity of swarm is the also highest among three patterns. Since it needs to enumerate every possible combination of objects, the time complexity is $O(2^n)$, where n is the number of moving objects in dataset. But by applying pruning rules on the search algorithm [21], swarm pattern mining is quite efficient in real scenario.

5.3 Trajectory Clustering

Different from moving object clusters that detect clusters of objects and the corresponding time intervals that they are being together, trajectory clustering will group (sub-)trajectories based on the overall trajectory similarity. Moving object cluster mining is more suitable to answer questions such as "find a group of people staying together for more than 2 hours", whereas trajectory clustering can answer questions like "group hurricane paths over years based on the trajectory similarity". Figure 12.11 illustrates an example of trajectory clustering. There are two clusters based on trajectory similarity.

A typical clustering framework needs to consider two factors: (1) similarity measure and (2) clustering methods. As we discuss earlier in Sect. 4, the typical similarity measures between two trajectories include Euclidean distance, Dynamic Time Warping and Longest Common Subsequence. And typical clustering methods include K-Means, Hierarchical clustering and Gaussian Mixture Model.

Gaffney and Smyth [12] propose to cluster trajectories based on a probabilistic modeling of trajectories. In probabilistic clustering, we assume that the data are being generated in the following "generative" manner:

- An individual is drawn randomly from the population of interest.
- The individual has been assigned to cluster k with probability w_k, $sum_{k=1}^{K} w_k = 1$. These are the prior weights on the K clusters.

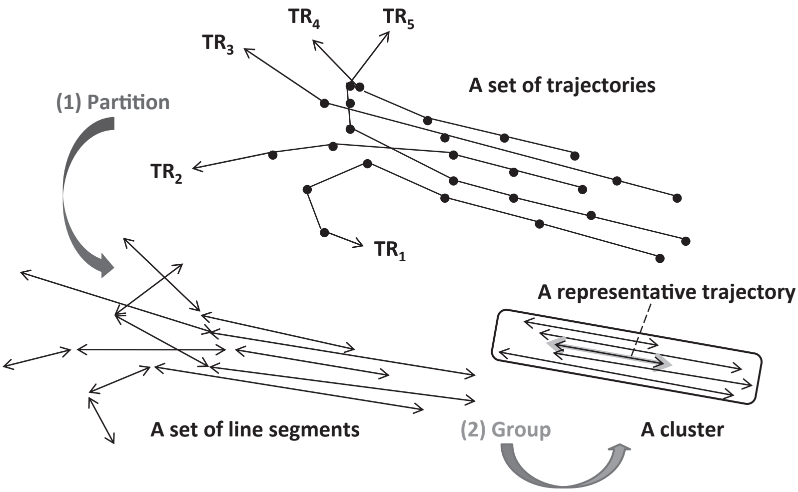

Fig. 12.12 An example of trajectory clustering in the partition-and-group framework [19]

- Given that an individual belongs to cluster k, there is a density function $f_k(y_j|\theta_k)$ which generates observed data y_i for individual j.

From this generative model, the observed density on the y's should be be a mixture model, i.e., a linear combination of the component models:

$$P(y_j|\theta) = \sum_k^K f_k(y_j|\theta_k)w_k.$$

In Gaussian mixture model, we will assume the generative models θ_k as Gaussian models. In Gaffney et al. [12], they assume the data is generated as mixtures of *regression models*, where we have measurements y which are a function of x and the density function becomes $f_k(y|x, \theta_k)$. Here x represents time, y represents the locations of object and θ_k is the regression model of y on x. The parameters in generative models can be estimated using the Expectation-Maximization (EM) algorithm. Experimental results [12] show that the proposed linear regression model performs slightly better than Gaussian mixture model. The difference becomes more obvious with higher standard deviation in data generation. Both mixture model methods perform much better than K-means.

In some applications, people are interested in discovering *similar portions* of trajectories. For example, meteorologists will be interested in the common behaviors of hurricanes near the coastline (i.e., at the time of landing) or at sea (i.e., before landing). To cluster sub-trajectories, Lee et al. [19] propose a partition-and-group framework named as TRACLUS as shown in Fig. 12.12. There are three steps in TRACLUS.

1. Partitioning: in this step, each trajectory is partitioned into a set of line segment based on characteristic points. A characteristic point is a point where the behavior

of a trajectory changes. The minimum description length (MDL) principle is adopted in this process.
2. Grouping: using given distance measure trajectory segments that are close to each other are grouped into a cluster. Density-based clustering algorithm is used in this process, which allows clusters in TRACULAS have any size and shape.
3. Representing: derive a representative trajectory for each cluster. The purpose of this representative trajectory is to describe the overall movement of the trajectory partitions that belong to the cluster.

An important step in TRACLUS is to partition a trajectory into sub-trajectories. By clustering sub-trajectories instead of the whole trajectories, we are able to discover the common paths shared by different sub-trajectories.

6 Summary

This chapter discusses many interesting state-of-the-art methods of spatiotemporal pattern mining. Discovery of spatiotemporal patterns can benefit various applications, such as ecological studies, traffic planning and social network analysis. We categorize the patterns as individual periodic patterns, pairwise patterns, and aggregate patterns over multiple trajectories.

As the collection of spatiotemporal data becomes easier and popular, spatiotemporal data mining is a promising research area with a lot of potential interesting research topics. There are still many challenging issues have not been well addressed by current methods, such as sparsity, uncertainties and noises in the data. It is also important to consider the spatial semantics (e.g., point of interest information) and constraints (e.g., road network and landscapes). So we could better understand the semantic meanings of the patterns. Finally, it will be interesting to consider human factor in the mining process and make the mining process more interactive and informative.

References

1. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 1995 Int. Conf. Data Engineering (ICDE'95)*, pages 3–14. IEEE, 1995.
2. M. Andersson, J. Gudmundsson, P. Laube, and T. Wolle. Reporting leaders and followers among trajectories of moving point objects. *GeoInformatica*, 12(4):497–528, 2008.
3. H. Cao, N. Mamoulis, and D. W. Cheung. Discovery of periodic patterns in spatiotemporal sequences. *Knowledge and Data Engineering, IEEE Transactions on*, 19(4):453–467, 2007.
4. L. Chen and R. T. Ng. On the marriage of lp-norms and edit distance. In *Proc. 2004 Int. Conf. Very Large Data Bases (VLDB'04)*, 2004.
5. L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proc. 2005 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'05)*, 2005.
6. J. Cranshaw, E. Toch, J. I. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proc. 2010 Int. Conf. Ubiquitous Computing (Ubicomp'10)*, 2010.

7. S. Dodge, R. Weibel, and A.-K. Lautenschütz. Towards a taxonomy of movement patterns. *Information visualization*, 7(3–4):240–252, 2008.
8. N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.
9. N. Eagle and A. S. Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.
10. N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. In *Proceedings of the National Academy of Sciences (PNAS'09)*, pages 15274–15278, 2009.
11. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. In *Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, Portland, OR, Aug. 1996.
12. S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining (KDD'99)*, pages 63–72, San Diego, CA, Aug. 1999.
13. F. Giannotti, M. Nanni, and D. Pedreschi. Efficient mining of temporally annotated sequences. In *Proc. 2006 SIAM Int. Conf. on Data Mining (SDM'06)*, 2006.
14. F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *Proc. 2007 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'07)*, pages 330–339. ACM, 2007.
15. J. Gudmundsson and M. van Kreveld. Computing longest duration flocks in spatio-temporal data. In *Proc. 2006 ACM Int. Symp. Advances in Geographic Information Systems (GIS'06)*, 2006.
16. J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In *Proc. 1999 Int. Conf. Data Engineering (ICDE'99)*, pages 106–115, Sydney, Australia, April 1999.
17. H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *Proc. 2008 Int. Conf. Data Engineering (ICDE'08)*, 2008.
18. H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of convoys in trajectory databases. In *Proc. 2008 Int. Conf. Very Large Data Bases (VLDB'08)*, 2008.
19. J.-G. Lee, J. Han, and K. Whang. Clustering trajectory data. In *Proc. 2007 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'07)*, Beijing, China, June 2007.
20. Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 34. ACM, 2008.
21. Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. In *Proc. 2010 Int. Conf. Very Large Data Bases (VLDB'10)*, Singapore, Sept. 2010.
22. Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *Proc. 2010 ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD'10)*, Washington D.C., July 2010.
23. Z. Li, C. X. Lin, B. Ding, and J. Han. Mining significant time intervals for relationship detection. In *Proc. 2011 Int. Symp. Spatial and Temporal Databases (SSTD'11)*, pages 386–403, 2011.
24. Z. Li, J. Wang, and J. Han. Mining periodicity for sparse and incomplete event data. In *Proc. of 2012 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'12)*, Beijing, China, Aug. 2012.
25. Z. Li, B. Ding, F. Wu, T. K. H. Lei, R. Kays, and M. C. Crofoot. Attraction and avoidance detection from movements. *Proceedings of the VLDB Endowment*, 7(3), 2013.
26. Z. Li, F. Wu, and M. C. Crofoot. Mining following relationships in movement data. In *Proc. 2013 Int. Conf. Data Mining (ICDM'13)*, 2013.
27. N. R. Lomb. Least-squares frequency analysis of unequally spaced data. In *Astrophysics and Space Science*, 1976.
28. N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proc. 2004 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'04)*, pages 236–245, Seattle, WA, Aug. 2004.

29. A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proc. 2009 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'09)*, pages 637–646, 2009.
30. J. M. Patel, Y. Chen, and V. P. Chakka. Stripes: An efficient index for predicted trajectories. In *Proc. 2004 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'04)*, Paris, France, June 2004.
31. S. Saltenis, C. Jensen, S. Leutenegger, and M. Lopez. Indexing the positions of continuously moving objects. In *Proc. 2003 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'03)*, pages 331–342, San Diego, CA, June 2003.
32. J. D. Scargle. Studies in astronomical time series analysis. ii - statistical aspects of spectral analysis of unevenly spaced data. In *Astrophysical Journal*, 1982.
33. T. F. Smith and M. S. Waterman. Comparison of biosequences. *Advances in Applied Mathematics*, 2(4):482–489, 1981.
34. Y. Tao and D. Papadias. Spatial queries in dynamic environments. *ACM Trans. Database Systems*, 28:101–139, 2003.
35. Y. Tao, D. Papadias, and J. Sun. The tpr*-tree: An optimized spatio-temporal access method for predictive queries. In *Proc. 2003 Int. Conf. Very Large Data Bases (VLDB'03)*, pages 790–801, Berlin, Germany, Sept. 2003.
36. Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proc. 2004 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'04)*, Paris, France, June 2004.
37. M. Vlachos, D. Gunopulos, and G. Kollios. Discovering similar multidimensional trajectories. In *Proc. 2002 Int. Conf. Data Engineering (ICDE'02)*, pages 673–684, San Francisco, CA, April 2002.
38. Y. Xia, Y. Tu, M. Atallah, and S. Prabhakar. Reducing data redundancy in location-based services. In *GeoSensor*, 2006.
39. B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proc. 1998 Int. Conf. Data Engineering (ICDE'98)*, pages 201–208, Orlando, FL, Feb. 1998.