Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data

Zhenhui Li and Jiawei Han

Abstract As spatiotemporal data becomes widely available, mining and understanding such data have gained a lot of attention recently. Among all important patterns, periodicity is arguably the most frequently happening one for moving objects. Finding periodic behaviors is essential to understanding the activities of objects, and to predict future movements and detect anomalies in trajectories. However, periodic behaviors in spatiotemporal data could be complicated, involving multiple interleaving periods, partial time span, and spatiotemporal noises and outliers. Even worse, due to the limitations of positioning technology or its various kinds of deployments, real movement data is often highly incomplete and sparse. In this chapter, we discuss existing techniques to mine periodic behaviors from spatiotemporal data, with a focus on tackling the aforementioned difficulties risen in real applications. In particular, we first review the traditional time-series method for periodicity detection. Then, a novel method specifically designed to mine periodic behaviors in spatiotemporal data, Periodica, is introduced. Periodica proposes to use reference spots to observe movement and detect periodicity from the in-and-out binary sequence. Then, we discuss the important issue of dealing with sparse and incomplete observations in spatiotemporal data, and propose a new general framework Periodo to detect periodicity for temporal events despite such nuisances. We provide experiment results on real movement data to verify the effectiveness of the proposed methods. While these techniques are developed in the context of spatiotemporal data mining, we believe that they are very general and could benefit researchers and practitioners from other related fields.

Zhenhui Li
Pennsylvania State University, University Park, PA, e-mail: jessieli@ist.psu.edu

Jiawei Han
University of Illinois at Urbana-Champaign, Champaign, IL, e-mail: hanj@cs.uiuc.edu

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. For example, understanding and modeling animal movement is important to addressing environmental challenges such as climate and land use change, bio-diversity loss, invasive species, and infectious diseases.

As spatiotemporal data becomes widely available, there are emergent needs in many applications to understand the increasingly large collections of data. Among all the patterns, one most common pattern 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. People may have weekly periodicity staying in the office.

Mining periodic behaviors can benefit us in many aspects. First, periodic behaviors provide an insightful and concise explanation over the long moving history. For example, animal movements can be summarized using mixture of multiple *daily* and *yearly* periodic behaviors. Second, periodic behaviors are also useful for compressing spatiotemporal data [17, 25, 4]. Spatiotemporal data usually have huge volume because data keeps growing as time passes. However, once we extract periodic patterns, it will save a lot of storage space by recording the periodic behaviors rather than original data, without losing much information. Finally, periodicity is extremely useful in future movement prediction [10], especially for a distant querying time. At the same time, if an object fails to follow regular periodic behaviors, it could be a signal of abnormal environment change or an accident.

More importantly, since spatiotemporal data is just a special class of temporal data, namely two-dimensional temporal data, many ideas and techniques we discuss in this chapter can actually be applied to other types of temporal data collected in a broad range of fields such as bioinformatics, social network, environmental science, and so on. For example, the notion of probabilistic periodic behavior can be very useful in understanding the social behaviors of people via analyzing the social network data such as tweets. Also, the techniques we developed for period detection from noisy and incomplete observations can be applied to any kind of temporal event data, regardless of the type of the collecting sensor.

1.1 Challenges in Mining Periodicity from Spatiotemporal Data

Mining periodic behaviors can bridge the gap between raw data and semantic understanding of the data, but it is a challenging problem. For example, Figure 1 shows the raw movement data of a student David along with the expected periodic behaviors. Based on manual examination of the raw data (on the left), it is almost impossible to extract the periodic behaviors (on the right). In fact, the periodic behaviors are quite complicated. There are multiple periods and periodic behaviors that may interleave with each other. Below we summarize the major challenges in mining periodic behavior from movement data:

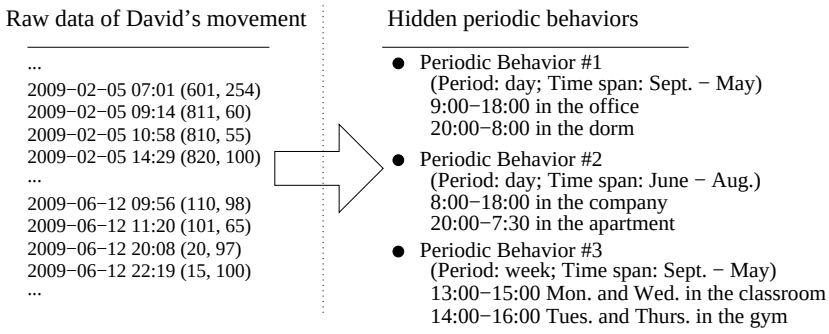


Fig. 1 Interleaving of multiple periodic behaviors

1. *A real life moving object does not ever strictly follow a given periodic pattern*. For example, birds never follow exactly the same migration paths 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 from raw trajectory data. 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.
2. We usually have *incomplete observations*, which *are unevenly sampled* and *have large portion of missing data*. For example, a bird can only carry small sensors with one or two reported locations in three to five days. And the locations of a person may only be recorded when he uses his cellphone. Moreover, if a sensor is not functioning or a tracking facility is turned off, it could result in a large portion of missing data.
3. With the periods detected, *the corresponding periodic behaviors should be mined* to provide a semantic understanding of movement data, such as the hidden periodic behaviors shown in Figure 1. The challenge in this step lies in the interleaving nature of multiple periodic behaviors. As we can see that, for a person's

movement as shown in Figure 1, one periodic behavior can be associated with different locations, such as periodic behavior #1 is associated with both office and dorm. Also, the same period (*i.e.*, day) could be associated with two different periodic behaviors, one from September to May and the other from June to August.

1.2 Existing Periodicity Mining Techniques

In this section, we will describe the existing periodicity mining techniques on various types of data, such as signal processing, gene data, and symbolic sequences. The techniques for spatiotemporal mining will be discussed in more detail in Section 2. Here we focus on two problems: (1) period detection and (2) periodic behavior mining. Period detection is to *automatically* detect the periods in time series or sequences. Periodic behavior mining problem is to mine periodic patterns with *a given period*.

1.2.1 Period Detection in Signals

A signal is a function that conveys information about the behavior or attributes of some phenomenon. If the function is on the time domain, the signal is a temporal function (*i.e.*, time series). The most frequently used method to detect periods in signals are *Fourier transform* and *autocorrelation* [18].

Fourier Transform maps a function of time into a new function whose argument is frequency with units of cycles/sec (hertz). In the case of a periodic function, the Fourier transform can be simplified to the calculation of a discrete set of complex amplitudes, called Fourier series coefficients. Given a sequence $x(n)$, $n = 0, 1, \ldots, N-1$, the normalized Discrete Fourier Transform is a sequence of complex numbers $X(f)$:

$$X(f_{k/N}) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}$$

where the subscript k/N denotes the frequency that each coefficient captures. In order to discover potential periodicities of a time series, one can use *periodogram* to estimate the spetral density of a signal. The periodogram P is provided by the squared length of each Fourier coefficient:

$$P(f_{k/N}) = \|X(f_{k/N})\|^2, k = 0, 1, \ldots, \lceil \frac{N-1}{2} \rceil$$

If $P(f_{k^*/N})$ is the maximum over all periodogram values of other frequencies, it means that frequency k^*/N has the strongest power in signal. Mapping frequency to time domain, a frequency k^*/N corresponds to time range $[\frac{N}{k^*}, \frac{N}{k^*-1})$.

Autocorrelation is the cross-correlation of a signal with itself. It is often used to find repeating patterns, such as the presence of a periodic signal. In statistics, autocorrelation of a time lag τ is defined as:

$$ACF(\tau) = \frac{1}{N}\sum_{n=0}^{N-1} x(\tau)\cdot x(n+\tau)$$

If $ACF(\tau^*)$ is the maximum over autocorrelation values of all time lags, it means that τ^* is most likely to be the period of the sequence. Different from Fourier transform that k^*/N is in frequency domain, time lag τ^* is in time domain.

Vlachos *et al.* [21] gives a comprehensive analysis and comparison between Fourier transform and autocorrelation. In general, Fourier transform is a great indicator for potential periods but the indicator is on the frequency domain. When mapping a frequency to time domain, it could correspond to a time range instead of one particular time. On the other hand, autocorrelation is not a good indicator for the true period because the true period and the multipliers of the true period will all have high autocorrelation values. For example, if τ^* is the true period, $ACF(k\cdot\tau^*)$ are all likely to have similar or even higher values than $ACF(\tau^*)$. Thus, it is hard to use a cut-off threshold to determine the true period. However, autocorrelation calculates the periodicity score on the time domain, so it does not have the mapping frequency problem in Fourier transform. In [21], Vlachos *et al.* proposes a method to combine autocorrelation and Fourier transform. It uses Fourier transform to find a good indicator of the potential period range and use autocorrelation to further validate the exact period.

1.2.2 Period Detection in Symbolic Sequences

Studies on period detection in data mining and database area usually assume the input to be a sequence of symbols instead of real value time series. A symbol could represent an event. An event could be a transaction record, for example, a person bought a bottle of milk. In transaction history, people could buy certain items periodically. Every timestamp is associated with one event or a set of events. The problem is to find whether there is an event or a set of events that have periodicity.

A common way to tackle the period detection in symbolic sequence is to get all the time indexes for each event and check whether these time indexes show periodicity. The time series that is being examined here can be considered as a binary sequence, $x = x_1x_2\ldots x_n$, where $x_t = 1$ means this event happens at time t and $x_t = 0$ means this event does not happen. The characteristics of such data is that the number of 1s could only be a very small portion in the sequence. And because of such sparsity, the period detection method is more sensitive to noise.

Ma *et al.* [16] proposes a chi-squared test for finding period by considering time differences in adjacent occurrences of an event. Let $s = \{t_1, t_2, \ldots, t_m\}$ denote all the timestamps that an event happens. It considers the time differences between every adjacent occurrences of the event: $\tau_i = t_{i+1} - t_i$. Looking at the histogram of all τ_i

values, the true period p should have high frequency. In this method, authors use Chi-square measure to set the threshold for the frequency. If a time difference value p has frequency more than this threshold, it outputs p as the period.

Berberdis *et al.* [3] uses autocorrelation to detect periods in the binary sequence x. Elfeky *et al.* [5] further improves this method by considering *multiple* events at the same time. It assumes that there is only one event at each timestamp. Each event is mapped to a binary sequence. For example, event "a" maps to "001", event "b" maps to "010", event "c" maps to "100". Then the original symbolic sequence input is transformed into a binary sequence. It further applies autocorrelation on this binary sequence to detect periods. In a follow-up work [5], Elfeky *et al.* mention the previous methods [3, 5] are sensitive to noises. These noises include insertion, deletion, replacement of an event at some timestamps. So [6] proposes a method based on Dynamic Time Warping to detect periods. The method is slower (*i.e.*, $O(n^2)$) compared with the previous method [5] (*i.e.*, $O(nlogn)$). But it is more accurate in terms of noises.

1.2.3 Period Detection in Gene Data

In bioinformatics, there are several studies in mining periods in gene data. A DNA sequence is a high-dimensional symbolic sequence. In [7], Glynn *et al.* mention that DNA sequence is often unevenly spaced and Fourier transform could fail when the data contains an excessive number of missing values. They propose to use Lomb-Scargle periodogram in such case. Lomb-Scargle periodogram [15, 19] is a variation of Fourier transform to handle unevenly spaced data using least-squares fitting of sinusoidal curves. In a follow-up work [1], Ahdesmäki *et al.* mention that Lomb-Scargle periodogram used in [7] is not robust since it is the basic Fisher's test. So they propose to use regression method for periodicity detection in non-uniformly sampled gene data. In [13], Liang *et al.* also mention that the performance of Lomb-Scargle periodogram [7] degrades in the presence of heavy-tailed non-Gaussian noise. In the presence of noises in gene data, Liang *et al.* [13] propose to use Laplace periodogram for more robust discovery of periodicity. They show Laplace periodogram is better than Lomb-Scarlge periodogram [7] and regression method [1]. An interesting previous study [11] has studied the problem of periodic pattern detection in sparse boolean sequences for gene data, where the ratio of the number of 1's to 0's is small. It proposes a scoring function for a potential period p by checking the alignment properties of periodic points in solenoidal coordinates w.r.t. p.

1.2.4 Periodic Behavior Mining

A number of *periodic pattern mining* techniques have been proposed in data mining literature. In this problem setting, each timestamp corresponds to a set of items. The goal is to, with a *given* period, find the period patterns that appear at least *min_sup*

times. Han *et al.* [9, 8] propose algorithms for mining frequent partial periodic patterns. Yang *et al.* [27, 28, 23, 29] propose a series of work dealing with variations of periodic pattern mining, such as asynchronous patterns [27], surprising periodic patterns [28], patterns with gap penalties [29], and higher level patterns [23]. In [30], it further addresses the gap requirement problem in biologic sequences. Different from previous works which focus on the categorical data, Mamoulis *et al.* [17] detects the periodic patterns for moving objects. Frequent periodic pattern mining tend to output a large set of patterns, most of which are slightly different.

1.3 Organization of this Chapter

In Section 2, we first review in more details the existing work on applying time-series methods to detect periodicity in spatiotemporal data. Then, we introduce a new approach, Periodica, which is able to discover complicated periodic behaviors from movement data. Section 3 is devoted to the important issue of detecting periodicity in real data: highly incomplete observations. We describe a novel method Periodo for robust periodicity detection for temporal events in these challenging cases, and verify its effectiveness by comparing it with existing methods on synthetic datasets. In Section 5, we show the results of applying the techniques introduced in this chapter to real spatiotemporal datasets, including the movement data of animals and humans. We conclude our discussion and point out future directions in Section 6.

2 Techniques for Periodicity Mining in Spatiotemporal Data

In this section, we describe techniques which are developed to detect periodic behaviors in spatiotemporal data. Let $D = \{(x_1, y_1, time_1), (x_2, y_2, time_2), \ldots\}$ be the original movement data for a moving object. Throughout this section, we assume that the raw data is linearly interpolated with constant time gap, such as hour or day. The interpolated sequence is denoted as $LOC = loc_1 loc_2 \cdots loc_n$, where loc_i is a spatial point represented as a pair $(loc_i.x, loc_i.y)$. Hence, our goal is to detect the periodicity in the movement sequence LOC.

While period detection in 1-D time series has been long studied, with standard techniques such as fast Fourier transform (FFT) and auto-correlation existing in the literature, solution to the problem of detecting periods in 2-D spatiotemporal data remains largely unknown until the recent work [2]. In this work, the authors first describe an intuitive approach to identify recursions in movement data, and then propose an extension of the 1-D Fourier Transform, named complex Fourier transform (CFT), to detect circular movements from the input sequence. Therefore, in this section we first review both methods, and point out their limitations in handling real-world movement data. Then, we show how such limitations can be overcome

using a novel two-stage algorithm, Periodica, which is designed to mine complex periodic behaviors from real-world movement data.

2.1 Existing Time-Series Methods

There have been many period detection methods developed for time series analysis. A direct usage of time series techniques requires we transform the location sequence into time series. A simple transform is mapping a location (x,y) onto complex plane $x+iy$, where $i=\sqrt{-1}$. We denote the mapping of a location loc_k as a complex number z_k, where $z_k = loc_k.x + iloc_k.y$.

2.1.1 Recursion Analysis

Recursion analysis is used to identify *closed paths* in the movement patterns. In order to define a closed path, or a recursion, one needs to divide the landscape into a grid of patches (a 105×105 matrix is used in [2]). Then, a close path exists in the movement sequence if an exact (to the resolution of landscape discretization) recursion to a previous location at a later time is found. To detect such recursions, one simply notices that the sum of vector displacements along a closed path is zero and thus requires the identification of zero-valued partial summations of the coordinates of sequential locations.

Specifically, given a sequence of locations vectors $z_k, k = 1,2,\ldots,n$, the method first compute the difference vectors $v_k = z_{k+1} - z_k$, for $k = 1,2,\ldots,n-1$. Then, for any time window $(s,t), t > s$, the segment of the path from z_s to z_t is denoted as $V(s,t)$:

$$V(s,t) = \sum_{k=s}^{t} v_k. \tag{1}$$

Thus, a recursion of duration D is a window for which $V(s,t) = 0$ and $t - s = D$. Notice that the recursion analysis identifies all closed paths, their length, and locations. These recursions are then sorted according to their durations to identify significant and semantic meaningful lengths of recursion (e.g., a day).

2.1.2 Circle Analysis

Fourier transform is one of the most widely used tools for time-series analysis. By extending it to complex numbers, one can identify circular paths, clockwise or counterclockwise, in the movement. Mathematically, given a sequence of location coordinates represented by a series of complex numbers $\{z_k\}_{k=1}^{n}$, the periodogram of the complex Fourier transform (CFT) of z_k is defined as:

$$Z(f) = \sum_{k=1}^{n} z_k \times e^{-i2\pi fk}, \quad f > 0 \tag{2}$$

Note that these spectra of Z are functions of the frequency f, which is the reciprocal of duration, D (*i.e.*, $D = 1/f$). It can be shown that $Z(f)$ provides an indication of the trend of circular motion, and can also be used to distinguish clockwise from counterclockwise patterns. Interested readers are referred to [2] for detailed illustrations and results of CFT.

Meanwhile, it is important to distinguish the circular analysis from the aforementioned recursion analysis. Note that a close path detected by recursion analysis is not necessarily circular, and similarly a clockwise or counterclockwise movement does not ensure a recursion. In this sense, these two methods are complementary to each other. Consequently, one can combine these two methods to answer more complex questions such as whether there is a circular path between recursions.

2.1.3 Limitations of Time-Series Methods

While tools from time-series analysis have demonstrated certain success when generalized to handle spatiotemporal data, it also has several major limitations as we elaborate below.

First, the performance of recursion analysis heavily rely on the resolution of landscape discretization, for which expert information about the moving objects' typical range of activity is crucial. For example, one will miss a lot of recursions when the resolution is set too coarse, whereas when the resolution is set too fine a large number of false positives will occur. Due to the same reason, the recursion analysis is also very sensitive to noise in the movement data.

Second, while circle analysis does not have the same dependency issue as recursion analysis, its usage is however strictly restricted to detecting circular paths in the movement data. Unfortunately, real-world spatiotemporal data often exhibit much more complex periodic patterns which are not necessarily circular (see Figure 2 for an example). Therefore, the development of a more flexible method is of great important in practice.

Finally, as we mentioned before, the objects of interest (e.g., humans, animals) often have multiple periodic behaviors with the same period, which is completely ignored by existing methods. In order to achieve semantic understanding of the data, it is important for our algorithm to be able to mine such multiple behaviors in movement data.

With all of these considerations in mind, we now proceed to describe a new algorithms for periodic behavior mining in spatiotemporal data, which handles all the aforementioned difficulties in a unified framework.

2.2 Periodica: Using Reference Spots to Detect Periodicity

As discussed above, periodic behaviors mined from spatiotemporal data can provide people with valuable semantic understanding of the movement. In order to mine periodic behaviors, one typically encounters the following two major issues.

First, the *periods* (*i.e.,* the regular time intervals in a periodic behavior) are usually unknown. Even though there are many period detection techniques that are proposed in signal processing area, such as Fourier transform and autocorrelation, we will see in Section 2.2.2 that these methods cannot be *directly* applied to the spatiotemporal data. Besides, there could be *multiple* periods existing at the same time, for example in Figure 1, David has one period as "day" and another as "week". If we consider the movement sequence as a whole, the longer period (*i.e.*, week) will have fewer repeating times than the shorter period (*i.e.*, day). So it is hard to select a threshold to find all periods. Surprisingly, there is no previous work that can handle the issue about how to detect multiple periods from the noisy moving object data.

Second, even if the periods are known, the *periodic behaviors* still need to be mined from the data because there could be *several* periodic behaviors with the same period. As we can see that, in David's movement, the same *period* (*i.e.*, day) is associated with two different *periodic behaviors*, one from September to May and the other from June to August. In previous work, Mamoulis *et al.* [17] studied the frequent periodic pattern mining problem for a moving object with a *given* period. However, the rigid definition of frequent periodic pattern does not encode the *statistical information*. It cannot describe the case such as "David has 0.8 probability to be in the office at 9:00 everyday." One may argue that these frequent periodic patterns can be further summarized using probabilistic modeling approach [26, 22]. But such models built on frequent periodic patterns do not truly reflect the real underlying periodic behaviors from the original movement, because frequent patterns are already a lossy summarization over the original data. Furthermore, if we can directly mine periodic behaviors on the original movement using polynomial time complexity, it is unnecessary to mine frequent periodic patterns and then summarize over these patterns.

We formulate the periodic behavior mining problem and propose the assumption that the observed movement is generated from several *periodic behaviors* associated with some *reference locations*. We design a two-stage algorithm, Periodica, to detect the periods and further find the periodic behaviors.

At the first stage, we focus on detecting all the periods in the movement. Given the raw data as shown in Figure 1, we use the kernel method to discover those reference locations, namely *reference spots*. For each reference spot, the movement data is transformed from a spatial sequence to a binary sequence, which facilitates the detection of periods by filtering the spatial noise. Besides, based on our assumption, every period will be associated with at least one reference spot. *All* periods in the movement can be detected if we try to detect the periods in every reference spot. At the second stage, we statistically model the periodic behavior using a *generative model*. Based on this model, underlying periodic behaviors are generalized from

the movement using a hierarchical clustering method and the number of periodic behaviors is automatically detected by measuring the *representation error*.

2.2.1 Problem Definition

Given a location sequence *LOC*, our problem aims at mining all periodic behaviors. Before defining periodic behavior, we first define some concepts. A *reference spot* is a dense area that is frequently visited in the movement. The set of all reference spots is denoted as $O = \{o_1, o_2, \ldots, o_d\}$, where d is the number of reference spots. A *period* T is a regular time interval in the (partial) movement. Let t_i $(1 \leq i \leq T)$ denote the i-th *relative timestamp* in T.

A *periodic behavior* can be represented as a pair $\langle T, \mathbf{P} \rangle$, where $\mathbf{P}$ is a probability distribution matrix. Each entry $\mathbf{P}_{ik}(1 \leq i \leq d, 1 \leq k \leq T)$ of $\mathbf{P}$ is the probability that the moving object is at the reference spot o_i at relative timestamp t_k.

As an example, for $T = 24$ (hours), David's daily periodic behavior (Figure 1 involved with 2 reference spots (*i.e.*, "office" and "dorm") could be represented as $(2+1) \times 24$ probability distribution matrix, as shown Table 1. This table is an intuitive explanation of formal output of periodic behaviors, which is not calculated according to specific data in Figure 1. The probability matrix encodes the noises and uncertainties in the movement. It statistically characterizes the periodic behavior such as "David arrives at office *around* 9:00."

	8:00	9:00	10:00	$\cdots$	17:00	18:00	19:00
dorm	0.9	0.2	0.1	$\cdots$	0.2	0.7	0.8
office	0.05	0.7	0.85	$\cdots$	0.75	0.2	0.1
unknown	0.05	0.1	0.05	$\cdots$	0.05	0.1	0.1

Table 1 A daily periodic behavior of David.

Definition 1 (Periodic Behavior Mining). Given a length-n movement sequence *LOC*, our goal is to mine all the periodic behaviors $\{\langle T, \mathbf{P} \rangle\}$.

Since there are two subtasks in the periodic behavior mining problem, detecting the periods and mining the periodic behaviors. We propose a two-stage algorithm Periodica, where the overall procedure of the algorithm is developed in two stages and each stage targets one subtask.

Algorithm 1 shows the general framework of Periodica. At the first stage, we first find all the reference spots (Line 2) and for each reference spot, the periods are detected (Lines 3∼5). Then for every period T, we consider the reference spots with period T and further mine the corresponding periodic behaviors (Lines 7∼10).

Algorithm 1 Periodica

INPUT: A movement sequence $LOC = loc_1 loc_2 \cdots loc_n$.
OUTPUT: A set of periodic behaviors.
ALGORITHM:

```
1: /* Stage 1: Detect periods */
2: Find reference spots O = {o_1, o_2, ..., o_d};
3: for each o_i ∈ O do
4:     Detect periods in o_i and store the periods in P_i;
5:     P_set ← P_set ∪ P_i;
6: end for
7: /* Stage 2: Mine periodic behaviors */
8: for each T ∈ P_set do
9:     O_T = {o_i | T ∈ P_i};
10:    Construct the symbolized sequence S using O_T;
11:    Mine periodic behaviors in S.
12: end for
```

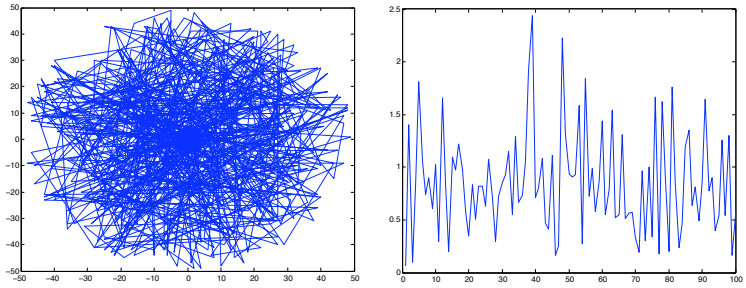
2.2.2 Detecting Period

In this section, we will discuss how to detect periods in the movement data. This includes two subproblems, namely, finding reference spots and detecting periods on binary sequence generated by these spots. First of all, we want to show why the idea of reference spots is essential for period detection. Consider the following example.

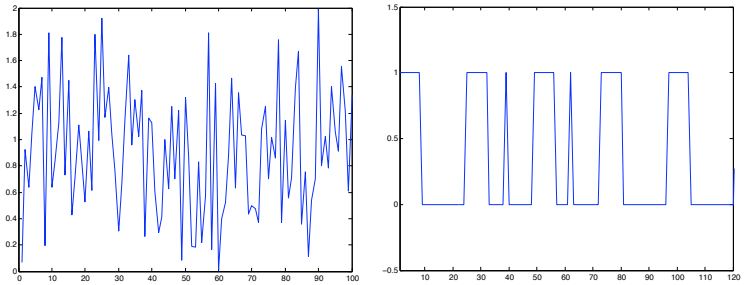
We generate a movement dataset simulating an animal's daily activities. Every day, this animal has 8 hours staying at the den and the rest time going to some random places hunting for food. Figure 2(a) shows its trajectories. We first try the method introduced in [2]. The method transforms locations (x,y) onto complex plane and use Fourier transform to detect the periods. However, as shown in Figure 2(b) and Figure 2(c), there is no strong signal corresponding to the correct period because such method is sensitive to the spatial noise. If the object does not follow more or less the same hunting *route* every day, the period can hardly be detected. However, in real cases, few objects repeat the exactly same route in the periodic movement.

Our key observation is that, if we view the data from the den, the period is easier to be detected. In Figure 2(d), we transform the movement into a binary sequence, where 1 represents the animal is at den and 0 when it goes out. It is easy to see the regularity in this binary sequence. Our idea is to find some important reference locations, namely *reference spots* , to view the movement. In this example, the den serves as our reference spot.

The notion of reference spots has several merits. First, it *filters out the spatial noise* and turns the period detection problem from a 2-dimensional space (*i.e.*, spatial) to a 1-dimensional space (*i.e.*, binary). As shown in Figure 2(d), we do not care where the animal goes when it is out of the den. As long as it follows a regular pattern going out and coming back to the den, there is a period associated with the den. Second, we can detect *multiple* periods in the movement. Consider the scenario that



(a) Raw trajectories (b) Fourier transform on $x+yi$



(c) Fourier transform on $y+xi$ (d) Binary sequence as viewed from the den

Fig. 2 Illustration of the importance to view movement from reference spots

there is a daily period with one reference spot and a weekly period with another reference spot, it is possible that only period "day" is discovered because the shorter period will repeat more times. But if we view the movement from two reference spots separately, both periods can be individually detected. Third, based on the assumption that each periodic behavior is associated with some reference locations, all the periods can be found through reference spots.

The rest of this section will discuss in details how to find reference spots and detect the periods on the binary sequence for each reference spot.

Finding Reference Spots. Since an object with periodic movement will repeatedly visit some specific places, if we only consider the spatial information of the movement, reference spots are those dense regions containing more points than the other regions. Note that the reference spots are obtained for each individual object.

Many methods can be applied to detect the reference spots, such as density-based clustering. The methods could vary according to different applications. We adapt a popular kernel method [24], which is designed for the purpose of finding home ranges of animals. For human movement, we may use important location detection methods in [14, 31].

While computing the density for each location in a continuous space is computationally expensive, we discretize the space into a regular $w \times h$ grid and compute the density for each cell. The grid size is determined by the desired resolution to view

the spatial data. If an animal has frequent activities at one place, this place will have higher probability to be its home. This actually aligns very well with our definition of reference spots.

For each grid cell c, the density is estimated using the bivariate normal density kernel,

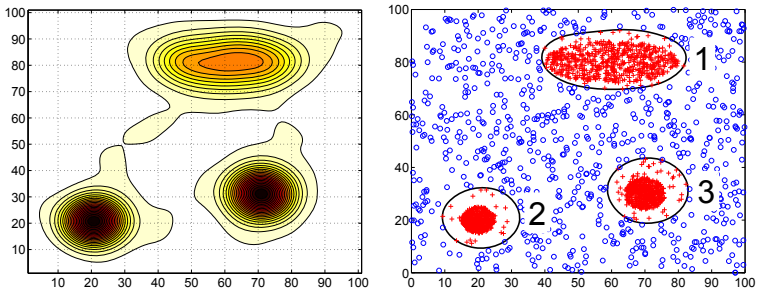
$$f(c) = \frac{1}{n\gamma^2} \sum_{i=1}^{n} \frac{1}{2\pi} \exp(-\frac{|c - loc_i|^2}{2\gamma^2}),$$

where $|c - loc_i|$ is the distance between cell c and location loc_i. In addition, γ is a smoothing parameter which is determined by the following heuristic method [2],

$$\gamma = \frac{1}{2}(\sigma_x^2 + \sigma_y^2)^{\frac{1}{2}} n^{-\frac{1}{6}},$$

where σ_x and σ_y are the standard deviations of the whole sequence LOC in its x and y-coordinates, respectively. The time complexity for this method is $O(w \cdot h \cdot n)$.

After obtaining the density values, a reference spot can be defined by a contour line on the map, which joins the cells of the equal density value, with some density threshold. The threshold can be determined as the top-p% density value among all the density values of all cells. The larger the value p is, the bigger the size of reference spot is. In practice, p can be chosen based on prior knowledge about the size of the reference spots. In many real applications, we can assume that the reference spots are usually very small on a large map (e.g., within 10% of whole area). So, by setting $p\% = 15\%$, most parts of reference spots should be detected with high probability.



(a) Density map calculated by kernel method. (b) Reference spots defined by contours.

Fig. 3 Finding reference spots.

To illustrate this idea, assume that a bird stays in a nest for half a year and moves to another nest staying for another half year. At each nest, it has a daily periodic behavior of going out for food during the daytime and coming back to the nest at night, as shown in Figure 3. Note that the two small areas (spot #2 and spot #3) are the two nests and the bigger region is the food resource (spot #1). Figure 3(a)

shows the density calculated using the kernel method. The grid size is 100×100. The darker the color is, the higher the density is. Figure 3(b) is the reference spots identified by contour using top-15% density value threshold.

Periods Detection on Binary Sequence. Given a set of reference spots, we further propose a method to obtain the potential periods within *each* spot *separately*. Viewed from a single reference spot, the movement sequence now can be transformed into a binary sequence $B = b_1 b_2 \dots b_n$, where $b_i = 1$ when this object is within the reference spot at timestamp i and 0 otherwise. In discrete signal processing area, to detect periods in a sequence, the most popular methods are Fourier transform and autocorrelation, which essentially complement each other in the following sense, as discussed in [21]. On one hand, Fourier transform often suffers from the low resolution problem in the low frequency region, hence provides poor estimation of large periods. Also, the well-known spectral leakage problem of Fourier transform tends to generate a lot of false positives in the periodogram. On the other hand, autocorrelation offers accurate estimation for both short and large periods, but is more difficult to set the significance threshold for important periods. Consequently, [21] proposed to combine Fourier transform and autocorrelation to find periods. Here, we adapt this approach to find periods in the binary sequence B.

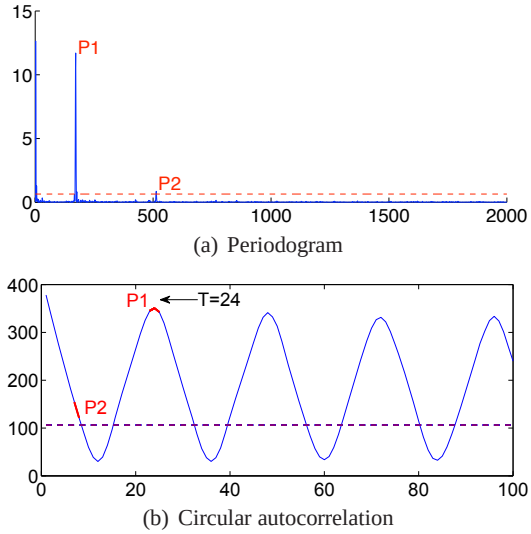
In Discrete Fourier Transform (DFT), the sequence $B = b_1 b_2 \dots b_n$ is transformed into the sequence of n complex numbers $X_1, X_2, \dots, X_n$. Given coefficients X, the periodogram is defined as the squared length of each Fourier coefficient: $F_k = \|X_k\|^2$. Here, F_k is the power of frequency k. In order to specify which frequencies are important, we need to set a threshold and identify those higher frequencies than this threshold.

The threshold is determined using the following method. Let B' be a randomly permutated sequence from B. Since B' should not exhibit any periodicities, even the maximum power does not indicate the period in the sequence. Therefore, we record its maximum power as p_{max}, and only the frequencies in B that have higher power than p_{max} may correspond to real periods. To provide a 99% confidence level on what frequencies are important, we repeat the above random permutation experiment 100 times and record the maximum power of each permutated sequence. The 99-th largest value of these 100 experiments will serve as a good estimator of the power threshold.

Given that F_k is larger than the power threshold, we still need to determine the exact period in the time domain, because a single value k in *frequency domain* corresponds to a range of periods $[\frac{n}{k}, \frac{n}{k-1})$ in *time domain*. In order to do this, we use circular autocorrelation, which examines how similar a sequence is to its previous values for different τ lags: $R(\tau) = \sum_{i=1}^{n} b_\tau b_{i+\tau}$.

Thus, for each period range $[l, r)$ given by the periodogram, we test whether there is a peak in $\{R(l), R(l+1), \dots, R(r-1)\}$ by fitting the data with a quadratic function. If the resulting function is concave in the period range, which indicates the existence of a peak, we return $t^* = \arg\max_{l \le t < r} R(t)$ as a detected period. Similarly, we employ a 99% confidence level to eliminate false positives caused by noise.

In Figure 4(a), we show the periodogram of reference spot #2 in Figure 3. The red dashed line denotes the threshold of 99% confidence. There are two points P_1



(a) Periodogram

(b) Circular autocorrelation

Fig. 4 Finding periods.

and P_2 that are above the threshold. In Figure 4(b), P_1 and P_2 are mapped to a range of periods. We can see that there is only one peak, P_1, corresponding to $T = 24$ on the autocorrelation curve.

2.2.3 Modeling Periodic Behaviors

After obtaining the periods for each reference spot, now we study the task how to mine periodic behaviors. We will consider the reference spots with the same period together in order to obtain more concise and informative periodic behaviors. But, since a behavior may only exist in a *partial* movement, there could be several periodic behaviors with the same period. For example, there are two daily behaviors in David's movement: One corresponds to the school days and the other occurs during the summer. However, given a long history of movement and a period as a "day", we actually do not know how many periodic behaviors exist in this movement and which days belong to which periodic behavior. This motivates us to use a clustering method. Because the "days" that belong to the same periodic behavior should have the similar temporal location pattern. We propose a generative model to measure the distance between two "days". Armed with such distance measure, we can further group the "days" into several clusters and each cluster represents one periodic behavior. As in David's example, "school days" should be grouped into one cluster and "summer days" should be grouped into another one. Note that, we assume that for each period, such as "day", one "day" will *only* belong to one behavior.

Since every period in the movement will be considered separately, *the rest of this section will focus on one specific period T*. First, we retrieve all the reference spots with period T. By combining the reference spots with the same period together, we will get a more informative periodic behaviors associated with different reference spots. For example, we can summarize David's daily behavior as "9:00∼18:00 at office *and* 20:00∼8:00 in the dorm". We do not consider combining two different periods in current work.

Let $O_T = \{o_1, o_2, \ldots, o_d\}$ denote reference spots with period T. For simplicity, we denote o_0 as any other locations outside the reference spots $o_1, o_2, \ldots, o_d$. Given $LOC = loc_1 loc_2 \cdots loc_n$, we generate the corresponding *symbolized movement sequence* $S = s_1 s_2 \ldots s_n$, where $s_i = j$ if loc_i is within o_j. S is further segmented into $m = \lfloor \frac{n}{T} \rfloor$ *segments*[1]. We use I^j to denote the j-th segment and t_k $(1 \leq k \leq T)$ to denote the k-th relative timestamp in a period. $I^j_k = i$ means that the object is within o_i at t_k in the j-th segment. For example, for $T = 24$ (hours), a segment represents a "day", t_9 denotes 9:00 in a day, and $I^5_9 = 2$ means that the object is within o_2 at 9:00 in the 5-th day. Naturally, we may use the categorical distribution to model the probability of such events.

Definition 2 (Categorical Distribution Matrix). Let $\mathscr{T} = \{t_1, t_2, \ldots, t_T\}$ be a set of relative timestamps, x_k be the categorical random variable indicating the selection of reference spot at timestamp t_k. $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_T]$ is a categorical distribution matrix with each column $\mathbf{p}_k = [p(x_k = 0), p(x_k = 1), \ldots, p(x_k = d)]^T$ being an independent categorical distribution vector satisfying $\sum_{i=0}^{d} p(x_k = i) = 1$.

Now, suppose I^1, I^2, …, I^l follow the same periodic behavior. The probability that the segment set $\mathscr{I} = \bigcup_{j=1}^{l} I^j$ is generated by some distribution matrix $\mathbf{P}$ is

$$P(\mathscr{I}|\mathbf{P}) = \prod_{I^j \in \mathscr{I}} \prod_{k=1}^{T} p(x_k = I^j_k).$$

Now, we formally define the concept of periodic behavior.

Definition 3 (Periodic Behavior). Let $\mathscr{I}$ be a set of segments. A periodic behavior over all the segments in $\mathscr{I}$, denoted as $\mathbf{H}(\mathscr{I})$, is a pair $\langle T, \mathbf{P} \rangle$. T is the period and $\mathbf{P}$ is a probability distribution matrix. We further let $|\mathscr{I}|$ denote the number of segments covered by this periodic behavior.

2.2.4 Discovery of Periodic Behaviors

With the definition of periodic behaviors, we are able to estimate periodic behaviors over a set of segments. Now given a set of segments $\{I^1, I^2, \ldots, I^m\}$, we need to discover which segments are generated by the same periodic behavior. Suppose there are K underlying periodic behaviors, each of which exists in a partial movement,

[1] If n is not a multiple of T, then the last ($n \mod T$) positions are truncated.

the segments should be partitioned into K groups so that each group represents one periodic behavior.

A potential solution to this problem is to apply some clustering methods. In order to do this, a distance measure between two periodic behaviors needs to be defined. Since a behavior is represented as a pair $\langle T, \mathbf{P} \rangle$ and T is fixed, the distance should be determined by their probability distribution matrices. Further, a small distance between two periodic behaviors should indicate that the segments contained in each behavior are likely to be generated from the same periodic behavior.

Several measures between the two probability distribution matrices $\mathbf{P}$ and $\mathbf{Q}$ can be used to fulfill these requirements. Here, since we assume the independence of variables across different timestamps, we propose to use the well-known Kullback-Leibler divergence as our distance measure:

$$KL(\mathbf{P}\|\mathbf{Q}) = \sum_{k=1}^{T} \sum_{i=0}^{d} p(x_k = i) \log \frac{p(x_k = i)}{q(x_k = i)}.$$

When $KL(\mathbf{P}\|\mathbf{Q})$ is small, it means that the two distribution matrices $\mathbf{P}$ and $\mathbf{Q}$ are similar, and vice versa.

Note that $KL(\mathbf{P}\|\mathbf{Q})$ becomes infinite when $p(x_k = i)$ or $q(x_k = i)$ has zero probability. To avoid this situation, we add to $p(x_k = i)$ (and $q(x_k = i)$) a background variable u which is uniformly distributed among all reference spots,

$$p(x_k = i) = (1 - \lambda) p(x_k = i) + \lambda u, \tag{3}$$

where λ is a small smoothing parameter $0 < \lambda < 1$.

Now, suppose we have two periodic behaviors, $\mathbf{H}_1 = \langle T, \mathbf{P} \rangle$ and $\mathbf{H}_2 = \langle T, \mathbf{Q} \rangle$. We define the distance between these two behaviors as

$$dist(\mathbf{H}_1, \mathbf{H}_2) = KL(\mathbf{P}\|\mathbf{Q}).$$

Suppose there exist K underlying periodic behaviors. There are many ways to group the segments into K clusters with the distance measure defined. However, the number of underlying periodic behaviors (*i.e.*, K) is usually unknown. So we propose a hierarchical agglomerative clustering method to group the segments while at the same time determine the optimal number of periodic behaviors. At each iteration of the hierarchical clustering, two clusters with the minimum distance are merged. In Algorithm 2, we first describe the clustering method assuming K is given. We will return to the problem of selecting optimal K later.

Algorithm 2 illustrates the hierarchical clustering method. It starts with m clusters (Line 1). A cluster C is defined as a collection of segments. At each iteration, two clusters with the minimum distance are merged (Lines 4∼8). When two clusters are merged, the new cluster inherits the segments that owned by the original clusters C_s and C_t. It has a newly built behavior $\mathbf{H}(C) = \langle T, \mathbf{P} \rangle$ over the merged segments, where $\mathbf{P}$ is computed by the following updating rule:

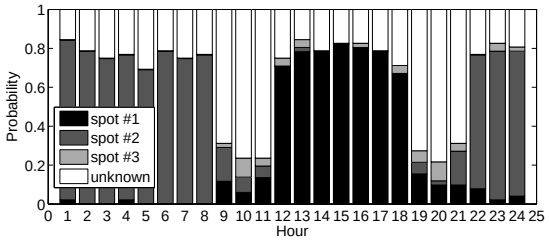
Algorithm 2 Mining periodic behaviors.

INPUT: symbolized sequence S, period T, number of clusters K.
OUTPUT: K periodic behaviors.
ALGORITHM:

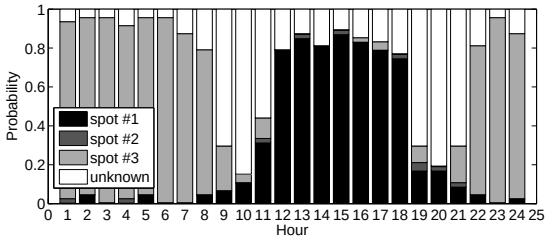
1: segment S into m segments;
2: initialize $k = m$ clusters, each of which has one segment;
3: compute the pairwise distances among $C_1, \ldots, C_k$, $d_{ij} = dist(\mathbf{H}(C_i), \mathbf{H}(C_j))$;
4: **while** $(k > K)$ **do**
5: select d_{st} such that $s,t = \arg\min_{i,j} d_{ij}$;
6: merge clusters C_s and C_t to a new cluster C;
7: calculate the distances between C and the remaining clusters;
8: $kv = k - 1$;
9: **end while**
10: return $\{\mathbf{H}(C_i), 1 \leq i \leq K\}$.

$$\mathbf{P} = \frac{|C_s|}{|C_s| + |C_t|}\mathbf{P}_s + \frac{|C_t|}{|C_s| + |C_t|}\mathbf{P}_t. \tag{4}$$

Finally, K periodic behaviors are returned (Line 9).



(a) **P** of periodic behavior #1



(b) **P** of periodic behavior #2

Fig. 5 Periodic behaviors.

To illustrate the method, we again use the example shown in Figure 3. There are two periodic behaviors with period $T = 24$ (hours) in the bird's movement. Figure 5 shows the probability distribution matrix for each discovered periodic behavior. A close look at Figure 5(a) shows that at time 0:00∼8:00 and 22:00∼24:00, the bird

has a high probability being at reference spot #2, which is a nest shown in Figure 3(b). At time 12:00∼18:00, it is very likely to be at reference spot #1, which is the food resources shown in Figure 3(b). And at the time 9:00∼11:00, there are also some probability that the bird is at reference spot #1 or reference spot #2. This indicates the bird goes out of the nest around 8:00 and arrives at the food resources place around 12:00. Such periodic behaviors well represent the bird's movement and truly reveal the mechanism we employed to generate this synthetic data.

Now, we discuss how to pick the appropriate parameter K. Ideally, during the hierarchical agglomerative clustering, the segments generated from the same behavior should be merged first because they have smaller KL-divergence distance. Thus, we judge a cluster is good if all the segments in the cluster are concentrated in one single reference spot at a particular timestamp. Hence, a natural representation error measure to evaluate the representation quality of a cluster is as follows. Note that here we exclude the reference spot o_0 which essentially means the location is unknown.

Definition 4 (Representation Error). Given a set of segments $C = \{I^1, I^2, \ldots, I^l\}$ and its periodic behavior $\mathbf{H}(C) = \langle T, \mathbf{P} \rangle$, the representation error is,

$$E(C) = \frac{\sum_{I^j \in C} \sum_{i=1}^{T} \mathbf{1}_{I_i^j \neq 0} \cdot (1 - p(x_i = I_i^j))}{\sum_{I^j \in C} \sum_{i=1}^{T} \mathbf{1}_{I_i^j \neq 0}}.$$

At each iteration, all the segments are partitioned into k clusters $\{C_1, C_2, \ldots, C_k\}$. The overall representation error at current iteration is calculated as the mean over all clusters,

$$\mathscr{E}_k = \frac{1}{k} \sum_{i=1}^{k} E(C_i).$$

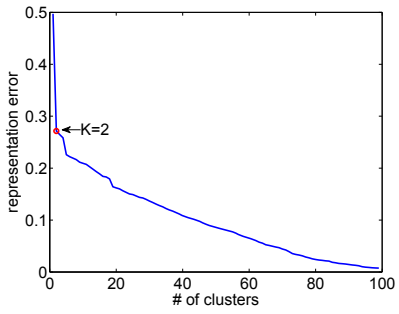


Fig. 6 Representation error.

During the clustering process, we monitor the change of $\mathscr{E}_k$. If $\mathscr{E}_k$ exhibits dramatical increases comparing with $\mathscr{E}_{k-1}$, it is a sign the newly merged cluster may contain two different behaviors and $k-1$ is likely to be a good choice of K. The

degree of such change can be observed from the derivative of $\mathscr{E}$ over k, $\frac{\partial \mathscr{E}}{\partial k}$. Since a sudden increase of $\mathscr{E}$ will result in a peak in its derivative, we can find the optimal K as $K = \arg\max_k \frac{\partial \mathscr{E}}{\partial k}$.

As we can see in Figure 6, the representation error suddenly increases at $k = 2$ for the bird's movement. This indicates that there are actually two periodic behaviors in the movement. This is true because the bird has one daily periodic behavior at the first nest and later has another one at the second nest.

3 Mining Periodicity from Incomplete Observations

So far, we have presented a complete framework, Periodica, for mining periodic behaviors from spatio-temporal data. Using the notion of reference spots, Periodica is able to discover complex periodic behaviors from real-world movement data. Nevertheless, we note that Periodica still relies on traditional periodicity analysis methods, namely Fourier transform and auto-correlation [18, 21, 5, 12], to detect periods after the movement data is converted to binary sequences. A fundamental assumption of all the traditional periodicity analysis methods is that they require the data to be *evenly sampled*, that is, there is an observation at every timestamp.

Unfortunately, due to the *limitations of data collection devices and methods*, this seemingly weak assumption is often seriously violated in practice. For example, a bird can only carry small sensors with one or two reported locations in three to five days. And the locations of a person may only be recorded when he uses his cellphone. Moreover, if a sensor is not functioning or a tracking facility is turned off, it could result in a large portion of missing data. Therefore, we usually have *incomplete observations*, which *are unevenly sampled* and *have large portion of missing data*. In fact, the issue with incomplete observations is a common problem on data collected from GPS and sensors, making period detection an even more challenging problem.

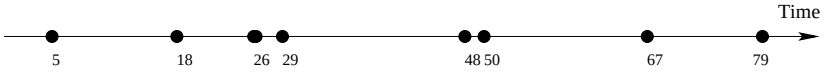


Fig. 7 Incomplete observations.

To illustrate the difficulties, let us first take a look at Figure 3. Suppose we have observed the occurrences of an event at timestamps 5, 18, 26, 29, 48, 50, 67, and 79. The observations of the event at other timestamps are not available. It is certainly not an easy task to infer the period directly from these *incomplete* observations. Even though some extensions of Fourier transform have been proposed to handle uneven data samples [15, 19], they are still not applicable to the case with very low sampling rate.

Besides, the periodic behaviors could be inherently *complicated and noisy*. A periodic event does not necessarily happen at *exactly* the same timestamp in each

periodic cycle. For example, the time that a person goes to work in the morning might *oscillate* between 8:00 to 10:00. *Noises* could also occur when the "in office" event is expected to be observed on a weekday but fails to happen.

In this section, we take a completely different approach to the period detection problem and handle all the aforementioned difficulties occurring in data collection process and periodic behavior complexity in a unified framework. The basic idea of our method is illustrated in Example 1.

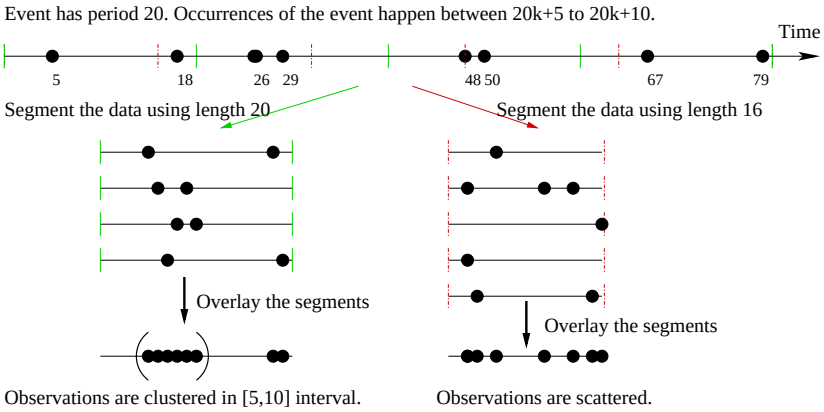


Fig. 8 Illustration example of our method.

Example 1. Suppose an event has a period $T = 20$ and we have eight observations of the event, as shown in Figure 3. If we overlay the observations with the correct period $T = 20$, we can see that most of the observations concentrate in time interval [5,10]. On the contrary, if we overlay the points with a wrong period, say $T = 16$, we cannot observe such clusters.

As suggested by Example 1, we could segment the timeline using a potential period T and summarize the observations over all the segments. If most of the observations fall into some time intervals, such as interval $[5, 10]$ in Example 1, T is *likely* to be the true period. In this section, we formally characterize such likelihood by introducing a probabilistic model for periodic behaviors. The model naturally handles the oscillation and noise issues because the occurrence of an event at any timestamp is now modeled with a probability. Next, we propose a new measure for periodicity based on this model. The measure essentially examines whether the distribution of observations is highly skewed w.r.t. a potential period T. As we will see later, even when the observations are incomplete, the overall distribution of observations, after overlaid with the correct T, remains skewed and is similar to the true periodic behavior model.

In summary, our major contributions are as follows. First, we introduce a probabilistic model for periodic behaviors and a random observation model for incom-

plete observations. This enables us to model all the variations we encounter in practice in a unified framework. Second, we propose a novel probabilistic measure for periodicity and design a practical algorithm to detect periods directly from the raw data. We further give rigorous proof of its validity under both the probabilistic periodic behavior model and the random observation model. Finally, we point out that our method can be used to detect periodicity for any temporal events, not necessarily restricting to movement data.

3.1 Problem Definition

Now we formally define the problem of period detection for events. We first assume that there is an observation at every timestamp. The case with incomplete observations will be discussed in Section 3.2.2. We use a binary sequence $\mathscr{X} = \{x(t)\}_{t=0}^{n-1}$ to denote observations. For example, if the event is "in the office", $x(t) = 1$ means this person is in the office at time t and $x(t) = 0$ means this person is *not* in the office at time t. Later we will refer $x(t) = 1$ as a *positive observation* and $x(t) = 0$ as a *negative observation*.

Definition 5 (Periodic Sequence). A sequence $\mathscr{X} = \{x(t)\}_{t=0}^{n-1}$ is said to be periodic if there exists some $T \in \mathbb{Z}$ such that $x(t+T) = x(t)$ for all values of t. We call T a period of $\mathscr{X}$.

A fundamental ambiguity with the above definition is that if T is a period of $\mathscr{X}$, then mT is also a period of $\mathscr{X}$ for any $m \in \mathbb{Z}$. A natural way to resolve this problem is to use the so called *prime period*.

Definition 6 (Prime Period). The prime period of a periodic sequence is the smallest $T \in \mathbb{Z}$ such that $x(t+T) = x(t)$ for all values of t.

For the rest of the section, unless otherwise stated, we always refer the word "period" to "prime period".

As we mentioned before, in real applications the observed sequences always deviate from the perfect periodicity due to the oscillating behavior and noises. To model such deviations, we introduce a new probabilistic framework, which is based on the *periodic distribution vectors* as defined below.

Definition 7 (Periodic Distribution Vector). For any vector $\mathbf{p}^T = [p_0^T, \ldots, p_{T-1}^T] \in [0,1]^T$ other than $\mathbf{0}^T$ and $\mathbf{1}^T$, we call it a periodic distribution vector of length T. A binary sequence $\mathscr{X}$ is said to be generated according to $\mathbf{p}^T$ if $x(t)$ is independently distributed according to Bernoulli$(p^T{}_{\bmod (t,T)})$.

Here we need to exclude the trivial cases where $\mathbf{p}^T = \mathbf{0}^T$ or $\mathbf{1}^T$. Also note that if we restrict the value of each p_i^T to $\{0,1\}$ only, then the resulting $\mathscr{X}$ is *strictly* periodic according to Definition 5. We are now able to formulate our period detection problem as follows.

Problem 1 (Event Period Detection). Given a binary sequence $\mathscr{X}$ generated according to any periodic distribution vector $\mathbf{p}^{T_0}$, find T_0.

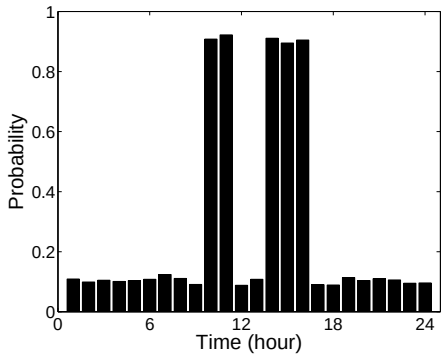


Fig. 9 (Running Example) Periodic distribution vector of a event with daily periodicity $T_0 = 24$.

Example 2 (Running Example). We will use a running example throughout the section to illustrate our method. Assume that a person has a daily periodicity visiting his office during 10am-11am and 2pm-4pm. His observation sequence is generated from the periodic distribution vector with high probabilities at time interval [10:11] and [14:16] and low but nonzero probabilities at other timestamps, as shown in Figure 9.

3.2 A Probabilistic Model For Period Detection

As we see in Example 3, when we overlay the binary sequence with its true period T_0, the resulting sequence correctly reveals its underlying periodic behavior. Now we make this observation formal using the concept of periodic distribution vector. Then, we propose a novel probabilistic measure of periodicity based on this observation and prove its validity even when observations are incomplete.

3.2.1 A Probabilistic Measure of Periodicity

Given a binary sequence $\mathscr{X}$, we define $S^+ = \{t : x(t) = 1\}$ and $S^- = \{t : x(t) = 0\}$ as the collections of timestamps with 1's and 0's, respectively. For a candidate period T, let $\mathscr{I}_T$ denote the power set of $[0 : T-1]$. Then, for any set of timestamps (*possibly non-consecutive*) $I \in \mathscr{I}_T$, we can define the collections of original timestamps that fall into this set after overlay as follows:

$$S_I^+ = \{t \in S^+ : \mathscr{F}_T(t) \in I\}, \quad S_I^- = \{t \in S^- : \mathscr{F}_T(t) \in I\},$$

where $\mathscr{F}_T(t) = \mod(t,T)$, and further compute the ratios of 1's and 0's whose corresponding timestamps fall into I after overlay:

$$\mu^+_{\mathscr{X}}(I,T) = \frac{|S^+_I|}{|S^+|}, \quad \mu^-_{\mathscr{X}}(I,T) = \frac{|S^-_I|}{|S^-|}. \tag{5}$$

The following lemma says that these ratios indeed reveal the true underlying probabilistic model parameters, given that the observation sequence is sufficiently long.

Lemma 1. *Suppose a binary sequence $\mathscr{X} = \{x(t)\}_{t=0}^{n-1}$ is generated according to some periodic distribution vector $\mathbf{p}^T$ of length T, write $q_i^T = 1 - p_i^T$. Then $\forall I \in \mathscr{I}_T$,*

$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}, \quad \lim_{n\to\infty} \mu^-_{\mathscr{X}}(I,T) = \frac{\sum_{i\in I} q_i^T}{\sum_{i=0}^{T-1} q_i^T}.$$

Proof. The proof is a straightforward application of the Law of Large Numbers (LLN), and we only prove the first equation. With a slight abuse of notation we write $S_i = \{t : \mathscr{F}_T(t) = i\}$ and $S_i^+ = \{t \in S^+ : \mathscr{F}_T(t) = i\}$. Since $\{x(t) : t \in S_i\}$ are i.i.d. Bernoulli(p_i^T) random variables, by LLN we have

$$\lim_{n\to\infty} \frac{|S_i^+|}{n} = \lim_{n\to\infty} \frac{\sum_{t\in S_i} x(t)}{n} = \frac{p_i^T}{T},$$

where we use $\lim_{n\to\infty} \frac{|S_i|}{n} = \frac{1}{T}$ for the last equality. So,

$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \lim_{n\to\infty} \frac{|S_I^+|/n}{|S^+|/n} = \lim_{n\to\infty} \frac{\sum_{i\in I} |S_i^+|/n}{\sum_{i=0}^{T-1} |S_i^+|/n} = \frac{\sum_{i\in I} p_i^T/T}{\sum_{i=0}^{T-1} p_i^T/T} = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}.$$

Now we introduce our measure of periodicity based on Lemma 1. For any $I \in \mathscr{I}_T$, its discrepancy score is defined as:

$$\Delta_{\mathscr{X}}(I,T) = \mu^+_{\mathscr{X}}(I,T) - \mu^-_{\mathscr{X}}(I,T). \tag{6}$$

Then, the periodicity measure of $\mathscr{X}$ w.r.t. period T is:

$$\gamma_{\mathscr{X}}(T) = \max_{I\in\mathscr{I}_T} \Delta(I,T). \tag{7}$$

It is obvious that $\gamma_{\mathscr{X}}(T)$ is bounded: $0 \le \gamma_{\mathscr{X}}(T) \le 1$. Moreover, $\gamma_{\mathscr{X}}(T) = 1$ if and only if $\mathscr{X}$ is strictly periodic with period T. But more importantly, we have the following lemma, which states that under our probabilistic periodic behavior model, $\gamma_{\mathscr{X}}(T)$ is indeed a desired measure of periodicity.

Lemma 2. *If a binary sequence $\mathscr{X}$ is generated according to any periodic distribution vector $\mathbf{p}^{T_0}$ for some T_0, then*

$$\lim_{n\to\infty} \gamma_{\mathscr{X}}(T) \le \lim_{n\to\infty} \gamma_{\mathscr{X}}(T_0), \quad \forall T \in \mathbb{Z}.$$

Proof. Define

$$c_i = \frac{p_i^{T_0}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{q_i^{T_0}}{\sum_{k=0}^{T_0-1} q_k^{T_0}},$$

it is easy to see that the value $\lim_{n\to\infty} \gamma_{\mathscr{Z}}(T_0)$ is achieved by $I^* = \{i \in [0, T_0-1] : c_i > 0\}$. So it suffices to show that for any $T \in \mathbb{Z}$ and $I \in \mathscr{I}_T$,

$$\lim_{n\to\infty} \Delta_{\mathscr{Z}}(I,T) \leq \lim_{n\to\infty} \Delta_{\mathscr{Z}}(I^*,T_0) = \sum_{i\in I^*} c_i.$$

Observe now that for any (I,T),

$$\lim_{n\to\infty} \mu^+_{\mathscr{Z}}(I,T) = \sum_{i\in I} \left(\frac{1}{T} \sum_{j=0}^{T_0-1} \frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} \right),$$

$$\lim_{n\to\infty} \mu^-_{\mathscr{Z}}(I,T) = \sum_{i\in I} \left(\frac{1}{T} \sum_{j=0}^{T_0-1} \frac{q^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} q_k^{T_0}} \right).$$

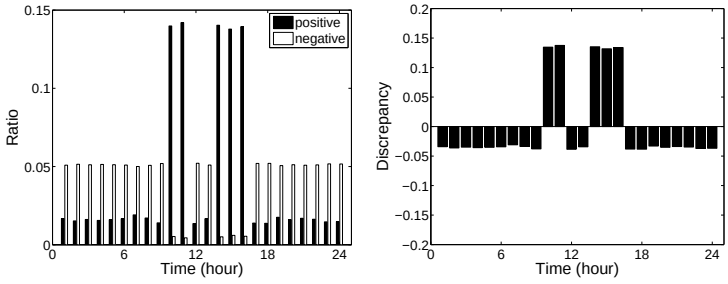
Therefore we have

$$\begin{aligned}
\lim_{n\to\infty} \Delta_{\mathscr{Z}}(I,T) &= \frac{1}{T} \sum_{i\in I} \sum_{j=0}^{T_0-1} \left(\frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{q^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} q_k^{T_0}} \right) \\
&= \frac{1}{T} \sum_{i\in I} \sum_{j=0}^{T_0-1} c_{\mathscr{F}_{T_0}(i+j\times T)} \leq \frac{1}{T} \sum_{i\in I} \sum_{j=0}^{T_0-1} \max(c_{\mathscr{F}_{T_0}(i+j\times T)}, 0) \\
&\leq \frac{1}{T} \sum_{j=0}^{T_0 T-1} \max(c_{\mathscr{F}_{T_0}(i+j\times T)}, 0) = \frac{1}{T} \times T \sum_{i\in I^*} c_i = \sum_{i\in I^*} c_i,
\end{aligned}$$

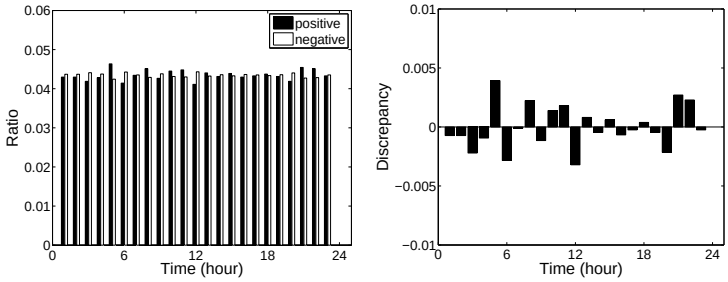
where the third equality uses the definition of I^*.

Note that, similar to the deterministic case, the ambiguity of multiple periods still exists as we can easily see that $\lim_{n\to\infty} \gamma_{\mathscr{Z}}(mT_0) = \lim_{n\to\infty} \gamma_{\mathscr{Z}}(T_0)$ for all m$\in \mathbb{Z}$. But we are only interested in finding the smallest one.

Example 3 (Running Example (cont.)). When we overlay the sequence using potential period $T = 24$, Figure 10(a) shows that positive observations have high probability to fall into the set of timestamps: $\{10, 11, 14, 15, 16\}$. However, when using the wrong period $T = 23$, the distribution is almost uniform over time, as shown in Figure 10(c). Similarly, we see large discrepancy scores for T = 24 (Figure 10(b)) whereas the discrepancy scores are very small for T = 23 (Figure 10(d)). Therefore, we will have $\gamma_{\mathscr{Z}}(24) > \gamma_{\mathscr{Z}}(23)$. Figure 11 shows the periodicity scores for all potential periods in $[1 : 200]$. We can see that the score is maximized at $T = 24$, which is the true period of the sequence.



(a) Ratios ($T = 24$) (b) Discrepancy ($T = 24$)



(c) Ratios ($T = 23$) (d) Discrepancy ($T = 23$)

Fig. 10 (a) and (c): Ratios of 1's and 0's at a single timestamp (*i.e.*, $\mu^{+}_{\mathscr{X}}(\cdot, T)$ and $\mu^{-}_{\mathscr{X}}(\cdot, T)$) when $T = 24$ and $T = 23$, respectively. (b) and (d): Discrepancy scores at a single timestamp (*i.e.* $\Delta_{\mathscr{X}}(\cdot, T)$) when $T = 24$ and $T = 23$.

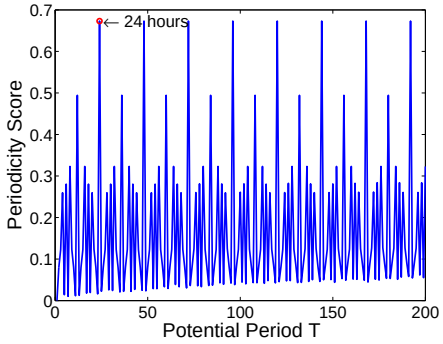


Fig. 11 Periodicity scores of potential periods.

3.2.2 Random Observation Model

Next, we extend our analysis on the proposed periodicity measure to the case of incomplete observations with a random observation model. To this end, we introduce a new label "-1" to the binary sequence $\mathscr{X}$ which indicates that the observation is unavailable at a specific timestamp. In the random observation model, each ob-

servation $x(t)$ is associated with a probability $d_t \in [0,1]$ and we write $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$.

Definition 8. A sequence $\mathscr{X}$ is said to be generated according to $(\mathbf{p}^T, \mathbf{d})$ if

$$x(t) = \begin{cases} \text{Bernoulli}(p^T_{\mathscr{F}_T(t)}) & \text{w.p. } d_t \\ -1 & \text{w.p. } 1-d_t \end{cases} \tag{8}$$

In general, we may assume that each d_t is independently drawn from some fixed but unknown distribution f over the interval $[0,1]$. To avoid the trivial case where $d_t \equiv 0$ for all t, we further assume that it has nonzero mean: $\rho_f > 0$. Although this model seems to be very flexible, in the section we prove that our periodicity measure is still valid. In order to do so, we need the following lemma, which states that $\mu^+_{\mathscr{X}}(I,T)$ and $\mu^-_{\mathscr{X}}(I,T)$ remain the same as before, assuming infinite length observation sequence.

Lemma 3. *Suppose* $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$ *are i.i.d. random variables in* $[0,1]$ *with nonzero mean, and a sequence* $\mathscr{X}$ *is generated according to* $(\mathbf{p}^T, \mathbf{d})$*, write* $q_i^T = 1 - p_i^T$*. Then* $\forall I \in \mathscr{I}_T$*,*

$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}, \quad \lim_{n\to\infty} \mu^-_{\mathscr{X}}(I,T) = \frac{\sum_{i\in I} q_i^T}{\sum_{i=0}^{T-1} q_i^T}.$$

Proof. We only prove the first equation. Let $y(t)$ be a random variable distributed according to Bernoulli(d_t) and $z(t) = x(t)y(t)$. Then $\{z(t)\}_{t=0}^{n-1}$ are independent random variables which take value in $\{0,1\}$, with mean $\mathbb{E}[z(t)]$ computed as follows:

$$\begin{aligned} \mathbb{E}[z(t)] &= \mathbf{P}(z(t)=1) = \mathbf{P}(x(t)=1, y(t)=1) \\ &= \mathbf{P}(x(t)=1 | y(t)=1)\mathbf{P}(y(t)=1) \\ &= p^T_{\mathscr{F}_T(t)}\mathbf{P}(y(t)=1) = p^T_{\mathscr{F}_T(t)}\mathbb{E}[d_t] = p^T_{\mathscr{F}_T(t)}\rho_f. \end{aligned}$$

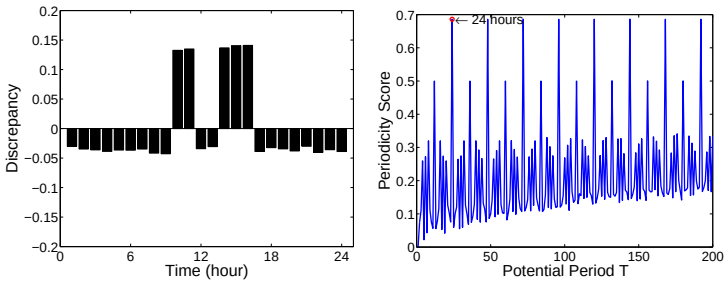
Define $S_i = \{t : \mathscr{F}_T(t) = i\}$ and $S_i^+ = \{t \in S^+ : \mathscr{F}_T(t) = i\}$, it is easy to see that $|S_i^+| = \sum_{t\in S_i} z(t)$. Using LLN we get

$$\lim_{n\to\infty} \frac{|S_i^+|}{n} = \lim_{n\to\infty} \frac{\sum_{t\in S_i} z(t)}{n} = \frac{p_i^T \rho_f}{T},$$

where we use $\lim_{n\to\infty} \frac{|S_i|}{n} = 1/T$ for the last equality. Therefore,

$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \lim_{n\to\infty} \frac{|S_I^+|/n}{|S^+|/n} = \lim_{n\to\infty} \frac{\sum_{i\in I} |S_i^+|/n}{\sum_{i=0}^{T-1} |S_i^+|/n} = \frac{\sum_{i\in I} \frac{p_i^T \rho_f}{T}}{\sum_{i=0}^{T-1} \frac{p_i^T \rho_f}{T}} = \frac{\sum_{i\in I} p_i^T}{\sum_{i=0}^{T-1} p_i^T}.$$

Since our periodicity measure only depends on $\mu^+_{\mathscr{X}}(I,T)$ and $\mu^-_{\mathscr{X}}(I,T)$, it is now straightforward to prove its validity under the random observation model. We summarize our main result as the following theorem.



(a) Discrepancy ($T = 24$) (b) Periodicity scores

Fig. 12 Period detection with unknown observations.

Theorem 1 *Suppose* $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$ *are i.i.d. random variables in* $[0,1]$ *with nonzero mean, and a sequence* $\mathscr{X}$ *is generated according to any* $(\mathbf{p}^{T_0}, \mathbf{d})$ *for some* T_0*, then*

$$\lim_{n\to\infty} \gamma_{\mathscr{X}}(T) \leq \lim_{n\to\infty} \gamma_{\mathscr{X}}(T_0), \quad \forall T \in \mathbb{Z}.$$

The proof is exactly the same as that of Lemma 2 given the result of Lemma 3, hence is omitted here.

Here we make two useful comments on this result. First, the assumption that d_t's are independent of each other plays an important role in the proof. In fact, if this does not hold, the observation sequence could exhibit very different periodic behavior from its underlying periodic distribution vector. But a thorough discussion on this issue is beyond the scope of this book. Second, this result only holds exactly with infinite length sequences. However, it provides a good estimate on the situation with finite length sequences, assuming that the sequences are long enough. Note that this length requirement is particularly important when a majority of samples are missing (*i.e.,* ρ_f is close to 0).

Example 4 (Running Example (cont.)). To introduce random observations, we sample the original sequence with sampling rate 0.2. The generated sequence will have 80% of its entries marked as unknown. Comparing Figure 12(a) with Figure 10(b), we can see very similar discrepancy scores over time. Random sampling has little effect on our period detection method. As shown in Figure 12(b), we can still detect the correct period at 24.

3.2.3 Handling Sequences Without Negative Samples

In many real world applications, negative samples may be completely unavailable to us. For example, if we have collected data from a local cellphone tower, we will know that a person is in town when he makes phone call through the local tower. However, we are not sure whether this person is in town or not for the rest of time because he could either be out of town or simply not making any call. In this case,

the observation sequence $\mathscr{X}$ takes value in $\{1,-1\}$ only, with -1 indicating the missing entries. In this section, we modify our measure of periodicity to handle this case.

Note that due to the lack of negative samples, $\mu^-_{\mathscr{X}}(I,T)$ can no longer be computed from $\mathscr{X}$. Thus, we need find another quantity to compare $\mu^+_{\mathscr{X}}(I,T)$ with. To this end, consider a binary sequence $\mathscr{U} = \{u(t)\}_{t=0}^{n-1}$ where each $u(t)$ is an i.i.d. Bermoulli(p) random variable for some fixed $p > 0$. It is easy to see that for any T and $I \in \mathscr{I}_T$, we have

$$\lim_{n\to\infty} \mu^+_{\mathscr{U}}(I,T) = \frac{|I|}{T}. \tag{9}$$

This corresponds to the case where the positive samples are evenly distributed over all entries after overlay. So we propose the new discrepancy score of I as follows:

$$\Delta^+_{\mathscr{X}}(I,T) = \mu^+_{\mathscr{X}}(I,T) - \frac{|I|}{T}, \tag{10}$$

and define the periodicity measure as:

$$\gamma^+_{\mathscr{X}}(T) = \max_{I\in\mathscr{I}_T} \Delta^+_{\mathscr{X}}(I,T). \tag{11}$$

In fact, with some slight modification to the proof of Lemma 2, we can show that it is a desired measure under our probabilistic model, resulting in the following theorem.

Theorem 2 *Suppose* $\mathbf{d} = \{d_t\}_{t=0}^{n-1}$ *are i.i.d. random variables in* $[0,1]$ *with nonzero mean, and a sequence* $\mathscr{X}$ *is generated according to any* $(\mathbf{p}^{T_0}, \mathbf{d})$ *for some* T_0*, then*

$$\lim_{n\to\infty} \gamma^+_{\mathscr{X}}(T) \le \lim_{n\to\infty} \gamma^+_{\mathscr{X}}(T_0), \quad \forall T \in \mathbb{Z}.$$

Proof. Define $c_i^+ = \frac{p_i^{T_0}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{1}{T_0}$, it is easy to see that the value $\lim_{n\to\infty} \gamma^+_{\mathscr{X}}(T_0)$ is achieved by $I^* = \{i \in [0, T_0-1] : c_i^+ > 0\}$. So it suffices to show that for any $T \in \mathbb{Z}$ and $I \in \mathscr{I}_T$,

$$\lim_{n\to\infty} \Delta^+_{\mathscr{X}}(I,T) \ \le \lim_{n\to\infty} \Delta^+_{\mathscr{X}}(I^*,T_0) = \sum_{i\in I^*} c_i^+.$$

Observe now that for any (I,T),

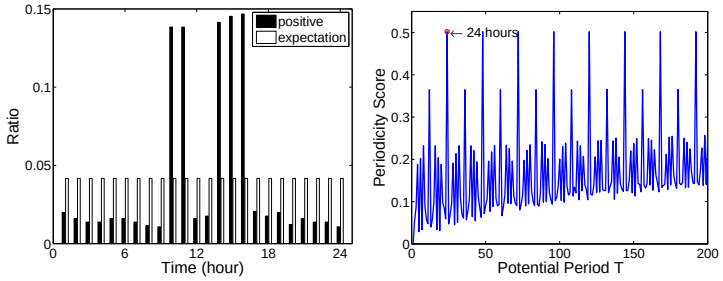
$$\lim_{n\to\infty} \mu^+_{\mathscr{X}}(I,T) = \sum_{i\in I} \left(\frac{1}{T} \sum_{j=0}^{T_0-1} \frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} \right).$$

Therefore we have

$$\begin{aligned}
\lim_{n\to\infty} \Delta^{+}_{\mathscr{Z}}(I,T) &= \frac{1}{T}\sum_{i\in I}\left\{\sum_{j=0}^{T_0-1}\left(\frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} p_k^{T_0}}\right)-1\right\} \\
&= \frac{1}{T}\sum_{i\in I}\sum_{j=0}^{T_0-1}\left(\frac{p^{T_0}_{\mathscr{F}_{T_0}(i+j\times T)}}{\sum_{k=0}^{T_0-1} p_k^{T_0}}-\frac{1}{T_0}\right) = \frac{1}{T}\sum_{i\in I}\sum_{j=0}^{T_0-1} c^{+}_{\mathscr{F}_{T_0}(i+j\times T)} \\
&\le \frac{1}{T}\sum_{i\in I}\sum_{j=0}^{T_0-1}\max(c^{+}_{\mathscr{F}_{T_0}(i+j\times T)},0) \le \frac{1}{T}\sum_{j=0}^{T_0T-1}\max(c^{+}_{\mathscr{F}_{T_0}(i+j\times T)},0) \\
&= \frac{1}{T}\times T\sum_{i\in I^*} c_i^{+} = \sum_{i\in I^*} c_i^{+},
\end{aligned}$$

where the fourth equality uses the definition of I^*.

Note that this new measure $\gamma^{+}_{\mathscr{Z}}(T)$ can also be applied to the cases where negative samples are available. Given the same validity result, readers may wonder if it can replace $\gamma_{\mathscr{Z}}(T)$. This is certainly not the case in practice, as our results only hold exactly when the sequence has infinite length. As we will see in experiment results, negative samples indeed provide additional information for period detection in finite length observation sequences.



(a) Distribution ($T = 24$) (b) Periodicity scores

Fig. 13 (Running Example) Period detection on sequences without negative samples.

Example 5 (Running Example (cont.)). In this example we further marked all the negative samples in the sequence we used in Example 4 as unknown. When there is no negative samples, the portion of positive samples at a single timestamp i is expected to be $\frac{1}{T}$, as shown in Figure 13(a). The discrepancy scores when $T = 24$ still have large values at $\{10, 11, 14, 15, 16\}$. Thus the correct period can be successfully detected as shown in Figure 13(b).

4 Algorithm: Periodo

In Section 3.2, we have introduced our periodicity measure for any potential period $T \in \mathbb{Z}$. Our period detection method simply computes the periodicity scores for every T and report the one with the highest score.

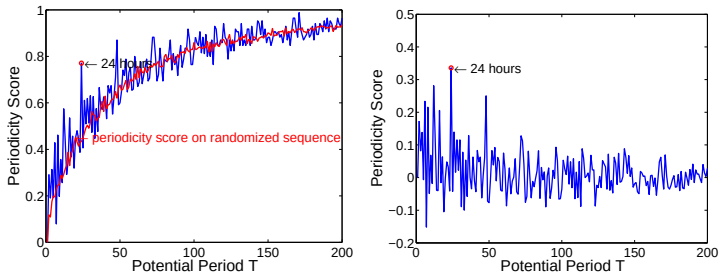
In this section, we first describe how to compute the periodicity score for a potential period and then discuss a practical issue when applying our method to finite length sequence. We will focus on the case with both positive and negative observations. The case without negative observations can be solved in the same way.

As we have seen in Section 3.2.1, the set of timestamps I^* that maximizes $\gamma_{\mathscr{X}}(T)$ can be expressed as

$$I^* = \{i \in [0, T_0 - 1] : c_i > 0\}, \tag{12}$$

where $c_i = \frac{p_i^{T_0}}{\sum_{k=0}^{T_0-1} p_k^{T_0}} - \frac{q_i^{T_0}}{\sum_{k=0}^{T_0-1} q_k^{T_0}}$. Therefore, to find I^*, it suffices to compute c_i for each $i \in [0, T_0 - 1]$ and select those ones with $c_i > 0$.

Time Complexity Analysis. For every potential period T, it takes $O(n)$ time to compute discrepancy score for a single timestamp (*i.e.*, c_i) and then $O(T)$ time to compute periodicity $\gamma_{\mathscr{X}}(T)$. Since potential period should be in range $[1, n]$, the time complexity of our method is $O(n^2)$. In practice, it is usually unnecessary to try all the potential periods. For example, we may have common sense that the periods will be no larger than certain values. So we only need to try potential periods up to n_0, where $n_0 \ll n$. This will make our method efficient in practice with time complexity as $O(n \times n_0)$.



(a) Periodicity scores (b) Normalized scores

Fig. 14 Normalization of periodicity scores.

Now we want to point out a practical issue when applying our method on finite length sequence. As one may already notice in our running example, we usually see a general increasing trend of periodicity scores $\gamma_{\mathscr{X}}(T)$ and $\gamma^+_{\mathscr{X}}(T)$ for a larger potential period T. This trend becomes more dominating as the number of observations decreases. For example, the original running example has observations for 1000 days. If the observations are only for 20 days, our method may result in incorrect period detection result, as the case shown in Figure 14(a). In fact, this phenomenon

is expected and can be understood in the following way. Let us take $\gamma^+_{\mathscr{X}}(T)$ as an example. Given a sequence $\mathscr{X}$ with *finite number* of positive observations, it is easy to see that the size of I that maximizes $\gamma^+_{\mathscr{X}}(T)$ for any T is bounded above by the number of positive observations. Therefore the value $\frac{|I^*|}{T}$ always decreases as T increases, no matter whether or not T is a true period of $\mathscr{X}$.

To remedy this issue for finite length sequence, we use periodicity scores on *randomized* sequence to normalize the original periodicity scores. Specifically, we randomly permute the positions of observations along the timeline and compute the periodicity score for each potential period T. This procedure is repeated N times and the average periodicity scores over N trials are output as the base scores. The redline in Figure 14(a) shows the base scores generated from randomized sequences by setting $N = 10$, which agree well with the trend.

For every potential period T, we subtract the base score from the original periodicity score, resulting in the normalized periodicity score. Note that the normalized score also slightly favors shorter period, which helps us to avoid detecting duplicated periods (*i.e.*, multiples of the prime period).

4.1 Experiment Results on Synthetic Datasets

In order to test the effectiveness of our method under various scenarios, we first use synthetic datasets generated according to a set of parameter. We take the following steps to generate a synthetic test sequence SEQ.

Step 1. We first fix a period T, for example, $T = 24$. The periodic segment SEG is a boolean sequence of length T, with values -1 and 1 indicating negative and positive observations, respectively. For simplicity of presentation, we write $SEG = [s_1 : t_1, s_2 : t_2, \ldots]$ where $[s_i, t_i]$ denote the i-th interval of SEG whose entries are all set to 1.

Step 2. Periodic segment SEG is repeated for TN times to generate the complete observation sequence, denoted as standard sequence SEQ_{std}. SEQ_{std} has length $T \times TN$.

Step 3 (Random sampling η). We sample the standard sequence with sampling rate η. For any element in SEQ_{std}, we set its value to 0 (*i.e.*, unknown) with probability $(1-\eta)$.

Step 4 (Missing segments α). For any segment in standard segment SEQ_{std}, we set all the elements in that segment as 0 (*i.e.*, unknown) with probability $(1-\alpha)$.

Step 5 (Random noise β). For any remaining observation in SEQ_{std}, we reverse its original values (making -1 as 1 and 1 as -1) with probability β.

The input sequence SEQ has values -1, 0, and 1 indicating negative, unknown, and positive observations. In the case when negative samples are unavailable, all the -1 values will be set to 0. Note that here we set negative observations as -1 and unknown ones as 0, which is different from the description in Section 3.1. The reason is that the unknown entries are set as -1, in the presence of many missing entries, traditional methods such as Fourier transform will be dominated by missing

entries instead of actual observations. The purpose of such adjustment is to facilitate traditional methods and it has no effect on our method.

4.1.1 Methods for Comparison

We compare our method with the following methods, which are frequently used to detect periods in boolean sequence [11].
1. Fourier Transform (FFT**)**: The frequency with the highest spectral power from Fourier transform via FFT is converted into time domain and output as the result.
2. Auto-correlation and Fourier Transform (Auto**)**: We first compute the auto-correlation of the input sequence. Since the output of auto-correlation will have peaks at all the multiples of the true period, we further apply Fourier transform to it and report the period with the highest power.
3. Histogram and Fourier Transform (Histogram**)**: We calculate the distances between any two positive observations and build a histogram of the distances over all the pairs. Then we apply Fourier transform to the histogram and report the period with the highest power.

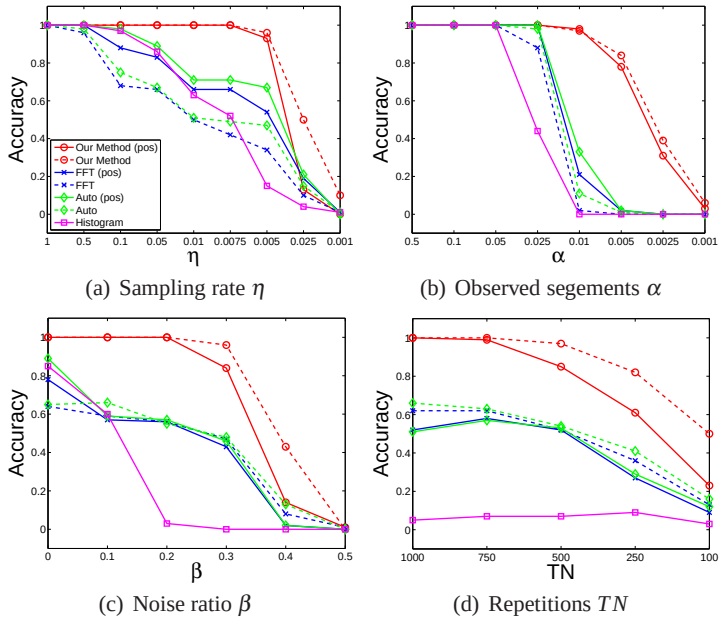
We will FFT(pos) and Auto(pos) to denote the methods FFT and Auto-correlation for cases without any negative observations. For Histogram, since it only considers the distances between positive observations, the results for cases with or without negative observations are exactly the same.

4.1.2 Performance Studies

In this section, we test all the methods on synthetic data under various settings. The default parameter setting is the following: $T = 24$, $SEG = [9 : 10, 14 : 16]$. $TN = 1000$, $\eta = 0.1$, $\alpha = 0.5$, and $\beta = 0.2$. For each experiment, we report the performance of all the methods with one of these parameters varying while the others are fixed. For each parameter setting, we repeat the experiment for 100 times and report the accuracy, which is the number of correct period detections over 100 trials. Results are shown in Figure 15.

Performance w.r.t. sampling rate η. To better study the effect of sampling rate, we set $\alpha = 1$ in this experiment. Figure 15(a) shows that our method is significantly better than other methods in terms of handling data with low sampling rate. The accuracy of our method remains 100% even when the sampling rate is as low as 0.0075. The accuracies of other methods start to decrease when sampling rate is lower than 0.5. Also note that Auto is slightly better than FFT because auto-correlation essentially generates a smoothed version of the categorical data for Fourier transform. In addition, it is interesting to see that FFT and Auto performs better in the case without negative observations.

Performance w.r.t. ratio of observed segments α. In this set of experiments, sampling rate η is set as 1 to better study the effect of α. Figure 15(b) depicts the performance of the methods. Our method again performs much better than other



(a) Sampling rate η (b) Observed segements α

(c) Noise ratio β (d) Repetitions TN

Fig. 15 Comparison results on synthetic data with various parameter settings.

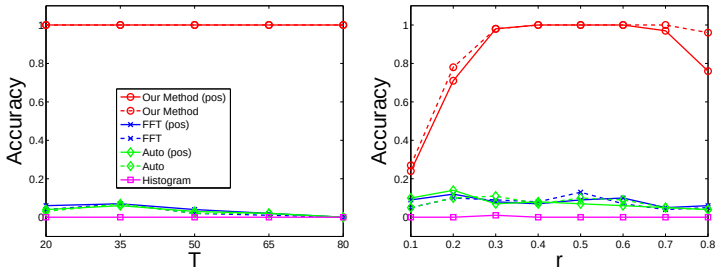
methods. Our method is almost perfect even when $\alpha = 0.025$. And when all other methods fail at $\alpha = 0.005$, our method still achieves 80% accuracy.

Performance w.r.t. noise ratio β. In Figure 15(c), we show the performance of the methods w.r.t. different noise ratios. Histogram is very sensitive to random noises since it considers the distances between any two positive observations. Our method is still the most robust one among all. For example, with $\beta = 0.3$, our method achieves accuracy as high as 80%.

Performance w.r.t. number of repetitions TN. Figure 15(d) shows the accuracies as a function of TN. As expected, the accuracies decrease as TN becomes smaller for all the methods, but our method again significantly outperforms the other ones.

Performance w.r.t. periodic behavior. We also study the performance of all the methods on randomly generated periodic behaviors. Given a period T and fix the ratio of 1's in a SEG as r, we generate SEG by setting each element to 1 with probability r. Sequences generated in this way will have positive observations scattered within a period, which will cause big problems for all the methods using Fourier transform, as evidenced in Figure 16. *This is because Fourier transform is very likely to have high spectral power at short periods if the input values alternate between 1 and 0 frequently.* In Figure 16(a) we set $r = 0.4$ and show the results w.r.t. period length T. In Figure 16(b), we fix $T = 24$ and show the results with varying r. As we can see, all the other methods fail miserably when the periodic behavior is

randomly generated. In addition, when the ratio of positive observations is low, *i.e.* fewer observations, it is more difficult to detect the correct period in general.



(a) True period T (b) Periodic segment SEG

Fig. 16 Comparison results on randomly generated periodic behaviors.

Parameter	Accuracy		
	Our Method	FFT	Lomb
$\eta = 0.5$	1	0.7	0.09
$\eta = 0.1$	1	0.52	0.10
$\alpha = 0.5$	1	1	0.01
$\alpha = 0.1$	0.99	0.35	0

Table 2 Comparison with Lomb-Scargle method.

Comparison with Lomb-Scargle method. Lomb-Scargle periodogram (Lomb) [15, 19] was introduced as a variation of Fourier transform to detect periods in *unevenly* sampled data. The method takes the timestamps with observations and their corresponding values as input. It does not work for the positive-sample-only case, because all the input values will be the same hence no period can be detected. The reason we do not compare with this method systematically is that the method performs poorly on the binary data and it is very slow. Here, we run it on a smaller dataset by setting $TN = 100$. We can see from Table 2 that, when $\eta = 0.5$ or $\alpha = 0.5$, our method and FFT perform well whereas the accuracy of Lomb is already approaching 0. As pointed out in [20], Lomb does not work well in bi-modal periodic signals and sinusoidal signals with non-Gaussian noises, hence not suitable for our purpose.

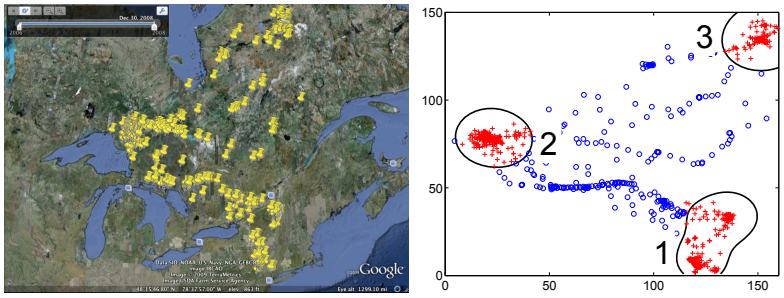
5 Experiments Results on Real Datasets

In this section, we demonstrate the effectiveness of the methods developed in this book on real-world spatio-temporal datasets. We first show the results of applying

our periodic behavior mining algorithm described in Section 2 to a real dataset of bald eagle movements[2]. This experiment verifies that the proposed method is able to discover semantic meaning periodic behaviors of real animals, as long as there are enough samples within each period. Then, we use real human movement data to test the new period detection method introduced in Section 3 when the observations are highly incomplete and unevenly sampled. The experiment results suggest that our method is extremely robust to uncertainties, noises and missing entries of the input data obtained in real-world applications.

5.1 Mining Periodic Behaviors: A bald Eagle Real Case

The data used in this experiment contains a 3-year tracking (2006.1∼2008.12) of a bald eagle in the North America. The data is first linearly interpolated using the sampling rate as a day.



(a) Raw data of bald eagle plotted on Google Earth.

(b) Reference spots.

Fig. 17 Real bald eagle data.

Figure 17(a) shows the original data of bald eagle using Google Earth. It is an enlarged area of Northeast in America and Quebec area in Canada. As shown in Figure 17(b), three reference spots are detected in areas of New York, Great Lakes and Quebec. By applying period detection to each reference spot, we obtain the periods for each reference spot, which are 363, 363 and 364 days, respectively. The periods can be roughly explained as a year. It is a sign of yearly migration in the movement.

Now we check the periodic behaviors mined from the movement. Ideally, we want to consider three reference spots together because they all show yearly period. However, we may discover that the periods are not exactly the same for all the reference spots. This is a very practical issue. In real cases, we can hardly get perfectly the same period for some reference spots. So, we should relax our constraint and

[2] The data set is obtained from www.movebank.org.

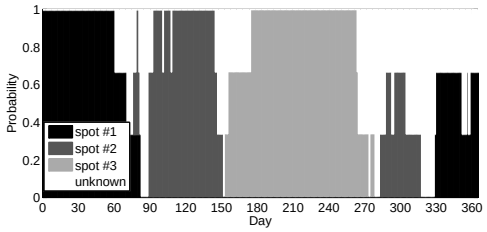


Fig. 18 Periodic behaviors of bald eagle.

consider the reference spots with *similar* periods together. If the difference of periods is within some tolerance threshold, we take the average of these periods and set it as the common period. Here, we take period T as 363 days, and the probability matrix is summarized in Figure 18. Using such probability matrix, we can well explain the yearly migration behavior as follows.

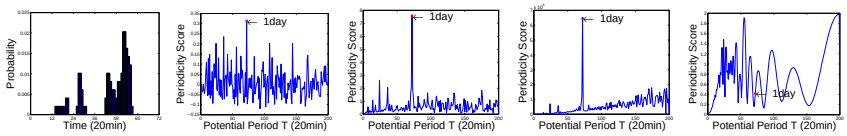
"*This bald eagle stays in New York area (i.e., reference spot # 1) from December to March. In March, it flies to Great Lakes area (i.e., reference spot #2) and stays there until the end of May. It flies to Quebec area (i.e., reference spot #3) in the summer and stays there until late September. Then it flies back to Great Lake again staying there from mid October to mid November and goes back to New York in December.*"

This real example shows the periodic behaviors mined from the movement provides an insightful explanation for the movement data.

5.2 Mining Periodicity from Incomplete Observations: Real Human Movements

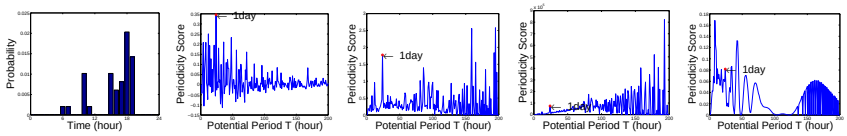
In this experiment, we use the real GPS locations of a person who has tracking record for 492 days. We first pick one of his frequently visited locations and generate a boolean observation sequence by treating all the visits to this location as positive observations and visits to other locations as negative observations. We study the performance of the methods on this symbolized movement data at different sampling rates. In Figure 19 and Figure 20, we compare the methods at two sampling rates, 20 minutes and 1 hour. As one can see in the figures (a) in Figure 19 and Figure 20, when overlaying this person's activity onto an period of one day, most of the visits occur in time interval [40, 60] for sampling rate of 20 minutes, or equivalently, in interval [15, 20] when the time unit is 1 hour. On one hand, when sampling rate is 20 minutes, all the methods except FFT(pos) and Histogram successfully detect the period of 24 hours, as they all have the strongest peaks at 24 hours (so we take 24 hours as the true period). On the other hand, when the data is sampled at each hour only, all the other methods fail to report 24 hours as the strongest peak whereas our method still succeeds. In fact, the success of our method can be easily inferred

from the left-most figures in Figure 19 and Figure 20, as one can see that lowering the sampling rate has little effect on the distribution graph of the overlaid sequence. We further show the periods reported by all the methods at various sampling rates in Table 3. Our method obviously outperforms the others in terms of tolerating low sampling rates.



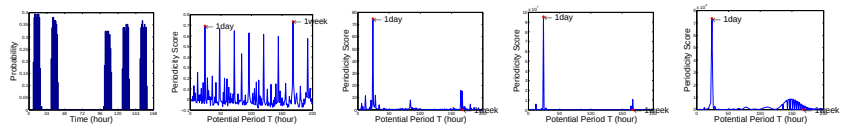
(a) Event (b) Our Method (c) FFT (d) Auto (e) Histogram

Fig. 19 [Sampling rate: 20 minutes] Comparison of period detection methods on a person's movement data.



(a) Event (b) Our Method (c) FFT (d) Auto (e) Histogram

Fig. 20 [Sampling rate: 1 hour] Comparison of period detection methods on a person's movement data.



(a) Event (b) Our Method (c) FFT (d) Auto (e) Histogram

Fig. 21 Comparion of methods on detecting long period, *i.e.* one week (168 hours).

Next, in Figure 21, we use the symbolized sequence of the same person at a different location and demonstrate the ability of our method in detecting multiple potential periods, especially those long ones. As we can see in Figure 21(a), this person clearly has weekly periodicity w.r.t. this location. It is very likely that this location is his office which he only visits during weekdays. Our method correctly detects 7-day with the highest periodicity score and 1-day has second highest score. But all other methods are dominated by the short period of 1-day. Please note that, in the figures of other methods, 1-week point is not even on the peak. This shows the strength of our method at detecting both long and short periods.

Method	Sampling rate			
	20min	1hour	2hour	4hour
Our Method (pos)	24	24	24	8
Our Method	24	24	24	8
FFT(pos)	9.3	9	8	8
FFT	24	195	372	372
Auto(pos)	24	9	42	8
Auto	24	193	372	780
Histogram	66.33	8	42	48

Table 3 Periods reported by different methods at various sampling rates.

6 Summary and Discussion

This chapter offers an overview of periodic pattern mining from spatiotemporal data. As movement data is widely available in larger volumes, the techniques of data mining nowadays play a crucial role in the semantic understanding and analysis of such data. The chapter first discusses the importance and challenges in mining periodic behaviors from movement data. We then review traditional time series methods for periodicity detection and discuss the disadvantages of directly applying these methods to movement data. To conquer these disadvantages, a novel approach, Periodica, is introduced. Periodica can detect multiple interleaved periodic behaviors from movement data by using the notion of reference spots. Next, we examine a common issue in real-world applications: the incomplete observations in spatiotemporal data. A robust period detection method for temporal events, Periodo, is then introduced to handle such sparse and incomplete movement data.

While experiment results on real movement data have already demonstrated the effectiveness of our methods, there are still many challenges that remain unsolved and new frontiers that would be interesting to explore. We list a few of them below.

First, in Periodica, there is a strong assumption that a reference spot must be a dense region on the map. However, a periodically visited place does not necessarily need to be dense in practice. For example, a person may go to Wal-Mart every Sunday afternoon. But compared with his home and office, Wal-Mart is not a densely visited location. If we use density-based method to find the reference spots, Wal-Mart is likely to be missed, even though this person has weekly periodic pattern with respect to it. Hence, designing a better method to identify such locations is a very interesting future direction.

Second, a more complicated yet more practical scenario in real data is the *irregular* periodic behavior. For example, the movement of fishing ships may follow the tides, which behave according to the cycles of the lunar phase. Hence, the movement of the ships may not have a strict monthly periodicity, which is defined based on the western calendar. Therefore, instead of simply saying "the ships roughly follow the monthly periodicity", it is desirable to develop new mechanisms which can explicitly model and detect such irregularity in the duration of a period.

Third, using periodic behaviors to predict future movements is a very important topic that deserves more in-depth study. Human and animals are highly dominated

by a mixture of their routines. For example, if we observe that a person is at home at 8am, how should we predict his location at 9am based on his routines? The correct answer may be the following. If it is a weekday, the next location should be the office; if it is a weekend, the next location could still be home; however, if it is a holiday, the next location might be somewhere on the way to his hometown. As we can see, the person's behavior is not confined to a single periodic behavior, but rather determined by multiple routines and the semantics of the locations and time. Therefore, it is very important to develop principled methodology that can fuse information from various sources to make reliable predictions.

7 Acknowledgments

The work was supported in part by Boeing company, NASA NRA-NNH10ZDA001N, NSF IIS-0905215 and IIS-1017362, the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA) and startup funding provided by the Pennsylvania State University. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

1. Miika Ahdesmäki, Harri Lähdesmäki, Andrew Gracey, and Olli Yli-Harja. Robust regression for periodicity detection in non-uniformly sampled time-course gene expression data. *BMC bioinformatics*, 8(1):233, 2007.
2. S. Bar-Dvaid, I. Bar-David, P. C. Cross, S. J. Ryan, and W. M. Getz. Methods for assessing movement path recursion with application to african buffalo in south africa. In *Ecology*, volume 90, 2009.
3. Christos Berberidis, Walid G. Aref, Mikhail J. Atallah, Ioannis P. Vlahavas, and Ahmed K. Elmagarmid. Multiple and partial periodicity mining in time series databases. In *Proc. 2002 European Conference on Artificial Intelligence (ECAI'02)*, 2002.
4. Huiping Cao, Nikos Mamoulis, and David W Cheung. Discovery of periodic patterns in spatiotemporal sequences. *Knowledge and Data Engineering, IEEE Transactions on*, 19(4):453–467, 2007.
5. Mohamed G. Elfeky, Walid G. Aref, and Ahmed K. Elmagarmid. Periodicity detection in time series databases. *IEEE Trans. Knowl. Data Eng.*, 17(7), 2005.
6. Mohamed G. Elfeky, Walid G. Aref, and Ahmed K. Elmagarmid. Warp: Time warping for periodicity detection. In *Proc. 2005 Int. Conf. Data Mining (ICDM'05)*, 2005.
7. Earl F. Glynn, Jie Chen, and Arcady R. Mushegian. Detecting periodic patterns in unevenly spaced gene expression time series using lomb-scargle periodograms. In *Bioinformatics*, 2005.
8. 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.
9. J. Han, W. Gong, and Y. Yin. Mining segment-wise periodic patterns in time-related databases. In *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98)*, pages 214–218, New York City, NY, Aug. 1998.

10. Hoyoung Jeung, Qing Liu, Heng Tao Shen, and Xiaofang Zhou. A hybrid prediction model for moving objects. In *Proc. 2008 Int. Conf. Data Engineering (ICDE'08)*, 2008.
11. Ivan Junier, Joan Herisson, and Francois Kepes. Periodic pattern detection in sparse boolean sequences. In *Algorithms for Molecular Biology*, 2010.
12. 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.
13. Kuo-ching Liang, Xiaodong Wang, and Ta-Hsin Li. Robust discovery of periodically expressed genes using the laplace periodogram. *BMC bioinformatics*, 10(1):15, 2009.
14. Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition using relational markov networks. In *Proc. 2005 Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, pages 773–778, 2005.
15. N. R. Lomb. Least-squares frequency analysis of unequally spaced data. In *Astrophysics and Space Science*, 1976.
16. S. Ma and J. L. Hellerstein. Mining partially periodic event patterns with unknown periods. In *Proc. 2001 Int. Conf. Data Engineering (ICDE'01)*, pages 205–214, Heidelberg, Germany, April 2001.
17. 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.
18. Maurice Bertram Priestley. *Spectral Analysis and Time Series*. London: Academic Press, 1981.
19. J. D. Scargle. Studies in astronomical time series analysis. ii - statistical aspects of spectral analysis of unevenly spaced data. In *Astrophysical Journal*, 1982.
20. Martin Schimmel. Emphasizing difficulties in the detection of rhythms with lomb-scargle periodograms. In *Biological Rhythm Research*, 2001.
21. Michail Vlachos, Philip S. Yu, and Vittorio Castelli. On periodicity detection and structural periodic similarity. In *Proc. 2005 SIAM Int. Conf. on Data Mining (SDM'05)*, 2005.
22. Chao Wang and Srinivasan Parthasarathy. Summarizing itemset patterns using probabilistic models. In *Proc. 2006 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'06)*, pages 730–735. ACM, 2006.
23. Wei Wang, Jiong Yang, and Philip S. Yu. Meta-patterns: Revealing hidden periodic patterns. In *Proc. 2001 Int. Conf. Data Mining (ICDM'01)*, San Jose, CA, Nov. 2001.
24. B. J. Worton. Kernel methods for estimating the utilization distribution in home-range studies. In *Ecology*, volume 70, 1989.
25. Yuni Xia, Yicheng Tu, Mikhail Atallah, and Sunil Prabhakar. Reducing data redundancy in location-based services. In *GeoSensor*, 2006.
26. X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: A profile-based approach. In *Proc. 2005 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'05)*, pages 314–323, Chicago, IL, Aug. 2005.
27. J. Yang, W. Wang, and P. S. Yu. Mining asynchronous periodic patterns in time series data. In *Proc. 2000 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'00)*, pages 275–279, Boston, MA, Aug. 2000.
28. J. Yang, W. Wang, and P. S. Yu. Infominer: mining surprising periodic patterns. In *Proc. 2001 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'01)*, pages 395–400, San Francisco, CA, Aug. 2001.
29. Jiong Yang, Wei Wang, and Philip S. Yu. Infominer+: Mining partial periodic patterns with gap penalties. In *Proc. 2002 Int. Conf. Data Mining (ICDM'02)*, Maebashi, Japan, Dec. 2002.
30. Minghua Zhang, Ben Kao, David Wai-Lok Cheung, and Kevin Y. Yip. Mining periodic patterns with gap requirement from sequences. In *Proc. 2005 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'05)*, pages 623–633, 2005.
31. Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World Wide Web (WWW'10)*, pages 1029–1038. ACM, 2010.