Mining Periodic Behaviors for Moving Objects

Zhenhui Li[†] Bolin Ding[†] Jiawei Han[†] Roland Kays[‡] Peter Nye[§]
[†] University of Illinois at Urbana-Champaign, Illinois, US
[‡] New York State Museum, New York, US
[§] New York State Department of Environmental Conservation, New York, US
{zli28, bding3, hanj}@uiuc.edu, rkays@mail.nysed.gov, fwinfo@gw.dec.state.ny.us

ABSTRACT

Periodicity is a frequently happening phenomenon for moving objects. Finding periodic behaviors is essential to understanding object movements. However, periodic behaviors could be complicated, involving multiple interleaving periods, partial time span, and spatiotemporal noises and outliers.

In this paper, we address the problem of mining periodic behaviors for moving objects. It involves two sub-problems: *how to detect the periods in complex movement*, and *how to mine periodic movement behaviors*. Our main assumption is that the observed movement is generated from multiple interleaved *periodic behaviors* associated with certain *reference locations*. Based on this assumption, we propose a two-stage algorithm, Periodica, to solve the problem. At the first stage, the notion of *reference spot* is proposed to capture the reference locations. Through reference spots, multiple periods in the movement can be retrieved using a method that combines Fourier transform and autocorrelation. At the second stage, a *probabilistic model* is proposed to characterize the periodic behaviors. For a specific period, periodic behaviors are statistically generalized from partial movement sequences through hierarchical clustering. Empirical studies on both synthetic and real data sets demonstrate the effectiveness of our method.

Categories and Subject Descriptors

H.2.8 [**Data Management**]: Database Aplications - Data Mining; H.4.0 [**Information Systems**]: General

General Terms

Algorithms

Keywords

Moving objects, periodic behavior, reference spot, Fourier Transform, autocorrelation, hierarchical clustering

KDD'10, July 25–28, 2010, Washington, DC, USA.

1. INTRODUCTION

With the fast development of positioning technology, massive amounts of object movement data have been collected from various moving object targets, such as animals, mobile devices, vehicles, and climate radars. As moving object data is widely available, mining and understanding such data has gained a lot of attention recently. One most common activity lying in moving objects is the *periodic behavior*. A periodic behavior can be loosely defined as the repeating activities at certain locations with regular time intervals. For example, golden eagles start migrating to South America in late October and go back to Alaska around mid March.

Such periodic behaviors provide an insightful and concise explanation over the long moving history. For example, animal movements could be summarized using several *daily* and *yearly* periodic behaviors. Periodic behaviors are also useful for compressing movement data [12, 17, 4]. Since they are summarization of movements, we can use them to replace the original data to save space. Moreover, periodic behaviors are 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.

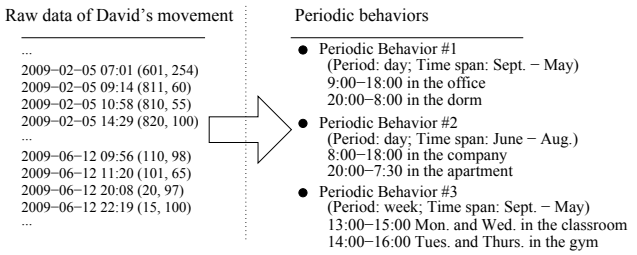


Figure 1: Interleaving of multiple periodic behaviors.

However, mining periodic behaviors from a moving object's long and noisy history data is a challenging problem. For example, Figure 1 shows the raw movement data of a student David and 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). And the periodic behaviors are actually quite complicated. There are multiple periods and periodic behaviors that may interleave with each other. Mining periodic behaviors can bridge the gap between raw data and semantic understanding of the data, which includes 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, these methods cannot be *directly* applied to the spatiotemporal data. Because the moving object will not repeat the movement by appearing at *exactly* the same point (in terms of (x, y)) on *exactly* the same time instance of a period. Besides, there could be *multiple* periods existing at the same time, such as 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. To the best of our knowledge, there is only one work [1] that addresses the detection of periods for moving objects. It directly applies the Fourier transform on moving object data by transforming a location onto a complex plane. However, as the toy example we will show in Section 3, this method does not work in the presence of spatial noise.

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.* [12] 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 [18, 14]. 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.

In this paper, 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*.

In summary, our major contributions are outlined as follows.

- We address an important problem in understanding movement data and formulate this problem as mining periodic behaviors.
- We propose algorithm Periodica to mine periodic behaviors. Periodica is designed in two stages.
- We design a location-based method to effectively detect multiple periods in the movement using the concept of reference spots.
- We statistically model the periodic behavior. A clustering method is proposed to determine the number of behaviors and mine periodic behaviors.
- Comprehensive experiments are conducted on both real data and complicated synthetic data. The results demonstrate the effectiveness of our method.

The rest of the paper is organized as follows. Section 2 formally states the problem and outlines the general framework. Section 3 introduces how to detect periods (stage 1). Section 4 describes the method to discover the periodic behaviors (stage 2). We report our experimental results in Section 5, discuss related work in Section 6, and conclude our study in Section 7.

2. FRAMEWORK OVERVIEW

Let $D = \{(x_1, y_1, time_1), (x_2, y_2, time_2), \ldots\}$ be the original movement database for a moving object. 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 $(loc_i.x, loc_i.y)$.

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. The formal statistical modeling of periodic behavior will be given in Section 4.1.

For 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 characterize the periodic behavior such as "David arrives at office *around* 9:00."

	8:00	9:00	10:00	···	17:00	18:00	19:00
dorm	0.9	0.2	0.1	···	0.2	0.7	0.8
office	0.05	0.7	0.95	···	0.75	0.2	0.1
unknown	0.05	0.1	0.05	···	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 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 (Section 3)*/
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: /* Stage 2: Mine periodic behaviors (Section 4) */
7: for each T ∈ P_set do
8:     O_T = {o_i | T ∈ P_i};
9:     Construct the symbolized sequence S using O_T;
10:    Mine periodic behaviors in S.
```

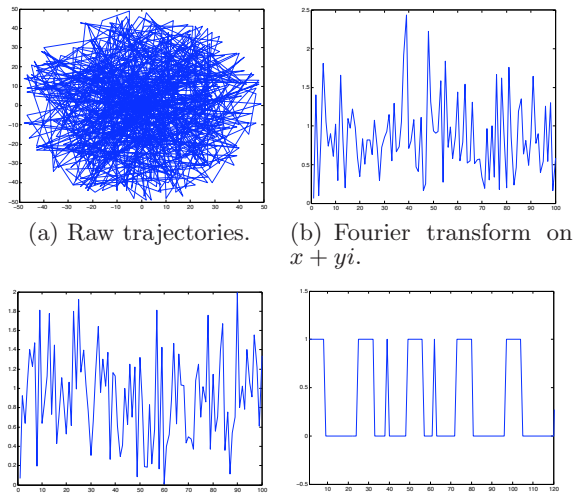
Algorithm 1 shows the general framework of Periodica. At the first stage, we first find all the reference spots (Line 2) and for each reference spot, the periods are detected (Line 3∼5). Then for every period T, we consider the reference spots with period T and further mine the corresponding periodic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

3. 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 [1]. 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



(a) Raw trajectories. (b) Fourier transform on $x + yi$. (c) Fourier transform on $y + xi$. (d) Binary sequence as viewed from the den.

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

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 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.

3.1 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 individual object. 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.

To estimate the density of each cell, we adapt a popu-

lar kernel method [16], which is designed for the purpose of finding home ranges of animals. 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 [16],

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

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 $O(w \cdot h \cdot n)$.

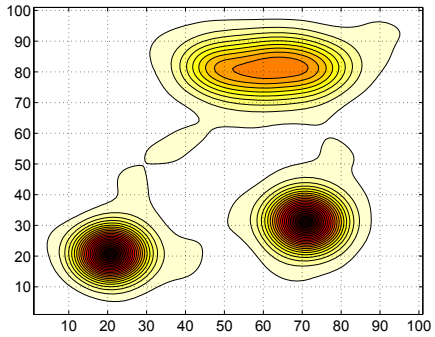
After obtaining the density values, an reference spot can be defined by a contour line on the map, which joins the cells of the equal density value, with some density threshold. The threshold can be determined as the top-p% density value among all the density values of all cells. The larger the value 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. Even though it could introduce a small amount of additional noise at the same time, our period detection is robust in terms of noise as shown in experiment, specifically in Figure 10.

EXAMPLE 1 (RUNNING EXAMPLE). *We will use a running example throughout the paper to illustrate our methods. 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.*

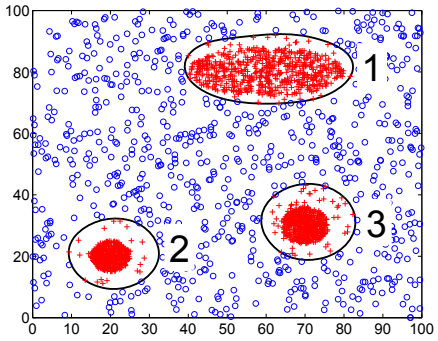
As shown in Figure 3, the two small areas (spot #2 and spot #3) are the two nests and the bigger region is the food resource (spot #1). Figure 3(a) shows the density calculated using the kernel method. The grid size is 100×100. *The darker the color is, the higher the density is. Figure 3(b) is the reference spots identified by contour using top-*15% *density value threshold.*

3.2 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 [13]. 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



(a) Density map calculated by kernel method.



(b) Reference spots defined by contours.

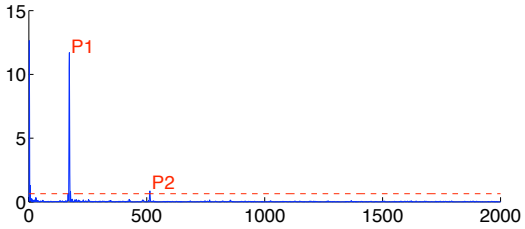
Figure 3: Finding reference spots.

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, [13] proposed to combine Fourier transform and autocorrelation to find periods. Here, we adapt this approach to find periods in the binary sequence B.

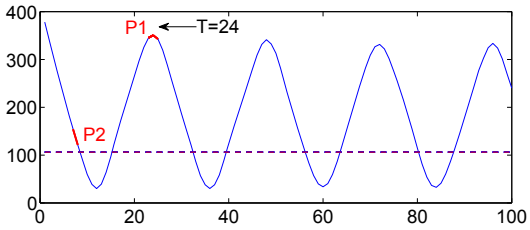
In Discrete Fourier Transform (DFT), the sequence $B = b_1 b_2 \dots b_n$ is transformed into the sequence of n complex numbers $X_1, X_2, \dots, X_n$. Given coefficients X, the periodogram is defined as the squared length of each Fourier coefficient: $F_k = \|X_k\|^2$. Here, F_k is the power of frequency 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



(a) Periodogram



(b) Circular autocorrelation

Figure 4: Finding periods.

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 $\{R(l), R(l+1), \ldots, R(r-1)\}$ 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.

EXAMPLE 2 (RUNNING EXAMPLE (CONT.)). *The periodogram of reference spot #2 is shown in Figure 4(a). The red dashed line denotes the threshold of 99% confidence. There are two points P_1 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. This suggests the existence of a period of 1 day in the movement data.*

Discrete Fourier Transform can be executed in $O(n \log n)$ time using Fast Fourier Transform algorithm (FFT). And since autocorrelation is a formal convolution which can also be solved by FFT, its complexity is also $O(n \log n)$. So, the overall time complexity of detecting periods in sequence B is $O(n \log n)$.

4. MINING 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 one 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.

In this section, we will formally present the technique to mine periodic behaviors. Since every period in the movement will be considered separately, *the rest of this section will focus on one specific period T.*

4.1 Modeling Periodic Behaviors

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 $(1 \le k \le T)$ 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 $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ 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 $\mathcal{I} = \bigcup_{j=1}^{l} I^j$ is generated by some distribution matrix $\mathbf{P}$ is

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

According to maximum likelihood estimation (MLE), the best generative model can be defined as the optimal solution to the following log likelihood maximization problem:

$$\max_{\mathbf{P}} \Big\{ L(\mathbf{P}|\mathcal{I}) = \log P(\mathcal{I}|\mathbf{P}) = \sum_{I^j \in \mathcal{I}} \sum_{k=1}^{T} p(x_k = I^j_k) \Big\}. \quad (1)$$

The well-known solution to (1) is

$$p(x_k = i) = \frac{\sum_{I^j \in \mathcal{I}} \mathbf{1}_{I^j_k = i}}{|\mathcal{I}|}, \quad (2)$$

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

where $\mathbf{1}_A$ is the indicator function associated with the event A. That is, $p(x_k = i)$ is the relative frequency of reference spot o_i at t_k over all segments in $\mathcal{I}$.

Now, we formally define the concept of periodic behavior.

DEFINITION 3 (Periodic Behavior). *Let $\mathcal{I}$ be a set of segments. A periodic behavior over all the segments in $\mathcal{I}$, denoted as $\mathbf{H}(\mathcal{I})$, is a pair $\langle T, \mathbf{P}\rangle$. T is the period and $\mathbf{P}$ is a probability distribution matrix learned through Eq.(2). We further let $|\mathcal{I}|$ denote the number of segments covered by this periodic behavior.*

4.2 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, 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 $0 < \lambda < 1$.

To further understand from a statistical point of view why this is a good choice of distance measure for our problem, let us return to our generative model. Recall that $\mathcal{I}$ is the set of segments generated by $\mathbf{P}$, then $KL(\mathbf{P}\|\mathbf{Q})$ can be decomposed as

$$\begin{aligned}
KL(\mathbf{P}\|\mathbf{Q}) &= \sum_{k=1}^{T}\sum_{i=0}^{d} p(x_k = i)\log p(x_k = i) \\
&\quad - \sum_{k=1}^{T}\sum_{i=0}^{d} p(x_k = i)\log q(x_k = i) \\
&= -H(\mathbf{P}) - \sum_{k=1}^{T}\sum_{i=0}^{d}\frac{\sum_{I^j\in\mathcal{I}}\mathbf{1}_{I_k^j=i}}{|\mathcal{I}|}\log q(x_k = i) \\
&= -H(\mathbf{P}) - \frac{1}{|\mathcal{I}|}\sum_{I^j\in\mathcal{I}}\sum_{k=1}^{T}\log q(x_k = I_k^j) \\
&= -H(\mathbf{P}) - \frac{1}{|\mathcal{I}|}\log P(\mathcal{I}|\mathbf{Q}),
\end{aligned}$$

where $H(\mathbf{P})$ is the entropy of $\mathbf{P}$ and can be regarded as a constant in our problem. Thus, the KL-divergence measures how likely the segment set $\mathcal{I}$ can be generated by the distribution matrix $\mathbf{Q}$. In our clustering algorithm, among all possible choices of $\mathbf{Q}$, we simply select the one that maximizes the likelihood $P(\mathcal{I}|\mathbf{Q})$.

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. We use a *representation error* to monitor the cluster quality. When the number of clusters turns from k to $k-1$, if the representation error increases dramatically, this indicates that k could be the correct number of periodic behaviors. We will first describe the clustering method as Algorithm 2 assuming K is given. The method to select optimal K is introduced in Section 4.3.

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: $k = k - 1$;
9: return $\{\mathbf{H}(C_i), 1 \le i \le K\}$.

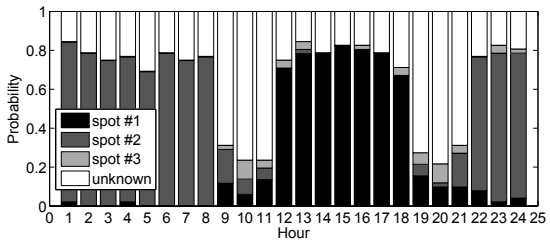
Algorithm 2 illustrates the hierarchical clustering method. It starts with m clusters (Line 1). A cluster C is defined as a

collection of segments. At each iteration, two clusters with the minimum distance are merged (Line 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:

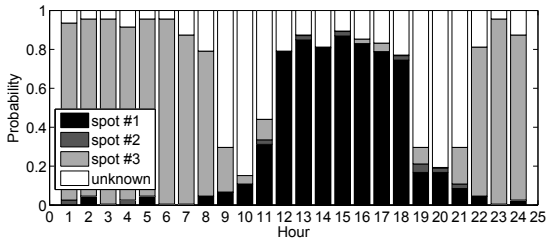
$$\mathbf{P} = \frac{|C_s|}{|C_s| + |C_t|}\mathbf{P}_s + \frac{|C_t|}{|C_s| + |C_t|}\mathbf{P}_t. \quad (4)$$

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

It takes $O(T \cdot d)$ to compute the distance between two behaviors, where d is the number of reference spots. The number of iterations is $O(m)$. At each iteration, it takes $O(m \log m)$ to find the minimum pair and $O(m \cdot T \cdot d)$ to compute the distances between the newly merged cluster with other clusters. In summary, the complexity of the clustering algorithm is $O(m \cdot (m \cdot T \cdot d + m \cdot \log m)) = O(m^2 \cdot T \cdot d + m^2 \cdot \log m)$.



(a) $\mathbf{P}$ of periodic behavior #1



(b) $\mathbf{P}$ of periodic behavior #2

Figure 5: Periodic behaviors.

EXAMPLE 3 (RUNNING EXAMPLE (CONT.)). *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.*

4.3 Number of Periodic Behaviors

In the clustering algorithm, K represents the number of periodic behaviors in the movement sequence. Since it is unknown how many periodic behaviors are in the movement, it is important to find the right way 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,

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

During the clustering process, we monitor the change of $\mathcal{E}_k$. If $\mathcal{E}_k$ exhibits a dramatical increases comparing with $\mathcal{E}_{k-1}$, 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 $\mathcal{E}$ over k, $\frac{\partial \mathcal{E}}{\partial k}$. Since a sudden increase of $\mathcal{E}$ will result in a peak in its derivative, we can find the optimal K as $K = \arg\max_k \frac{\partial \mathcal{E}}{\partial k}$.

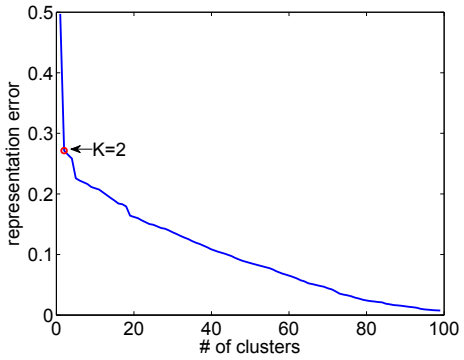


Figure 6: Representation error.

EXAMPLE 4 (RUNNING EXAMPLE (CONT.)). *As we can see Figure 6, the representation error suddenly increases at $k = 2$. 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.*

5. EXPERIMENT

In this section, we systematically evaluate the techniques presented in the paper. The language used is C++ and the

experiments are performed on a 2.8 GHz Intel Core 2 Duo system with 4GB memory. The system ran MAC OS X with version 10.5.5 and gcc 4.0.1.

In order to test the effectiveness under various scenarios, we design a generator for moving objects with periodicity according to a set of parameter values. These parameters are the length n of the time history (in timestamps), period T, the probability α for a periodic segment in the object's movement to comply with regular movement, the probability β for the noise for each timestamp in a regular periodic segment, and the variance σ of normal distribution to add temporal perturbations to the periodic segment.

Before generating the movement, we first create several reference spots. Each reference spot is a small circle with radius ranges from 1% to 5% of the map size. A standard segment seg_{std} with length T is the movement following the regular periodic pattern. For example, for $T = 24$ (hours), seg_{std} could be designed as 6:00pm∼8:00am at reference spot A (such as home) and 8:30am ∼ 5:30pm hours at reference spot B (such as office). Then, the movement of the object is generated. For every segment seg, we first determine whether s should be a regular segment or not, given the probability α.

If seg is a regular segment, the object's movement is generated as follows. According to standard segment, suppose that from timestamp t_0 to t_1 the object is at reference spot A, we further perturb t_0 and t_1 with some normal distribution (*i.e.*, $t_0' = N(t_0, \sigma^2)$, $t_1' = N(t_1, \sigma^2)$). For all the experiments, we fix $\sigma = 0.5$. Finally, with probability $1 - \beta$, the object is at a random location within the circle of reference spot A from t_0' to t_1'. For other timestamps that are not confined to any reference spot, a random location is generated. If seg is an irregular segment, for each timestamp, a random location is assigned.

5.1 Case Studies

We first conduct some case studies on both synthetic and real data sets.

5.1.1 A synthetic case with multiple periods

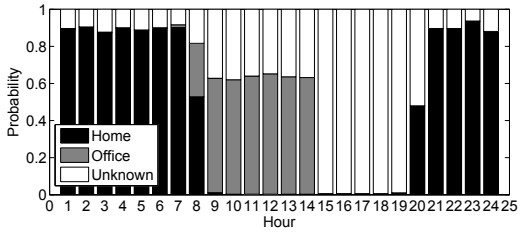
Since the running example has already illustrated periodic behaviors in partial movement, here we test our algorithm on a case with multiple periods. Suppose that there are 4 reference spots. Imagine them as "home", "office", "gym", and "class". A standard movement segment is generated as 20:00∼8:00 at home every day; 9:00∼14:00 at office on weekdays; 15:00∼17:00 at gym on Tuesdays and Thursdays; 15:00∼17:00 at class on Mondays, Wednesdays and Fridays. Furthermore, we choose $n = 8400$, $\alpha = 0.9$ and $\beta = 0.1$.

Obs. Spot	Home	Office	Gym	Class
Periods (hours)	24	24, 168	168	168

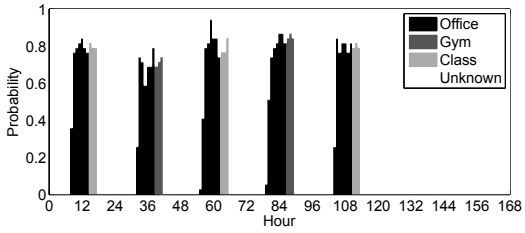
Table 2: Periods detected.

The periods detected for each reference spot are shown in Table 2. There are two periods detected: 24 (*i.e.*, day) and 168 (*i.e.*, week). It is interesting to see that office has both 24 and 168 as the periods. This is because office is visited "almost" every day except weekends. So both day and week are reasonable periods.

There is one daily behavior and one weekly behavior. Their probability matrices are illustrated in Figure 7. In Figure 7(a), we can infer that this person leaves home *around* 8:00am because the probability starts to drop at 8:00am. In the weekly movement shown in Figure 7(b), 9:00∼14:00 weekdays, the person stays in the office with high probability. Gym is involved with Tuesday and Thursday afternoons and class is involved with Monday, Wednesday and Friday afternoons. The behaviors on weekends are unknown.



(a) Periodic behavior for $T = 24$.

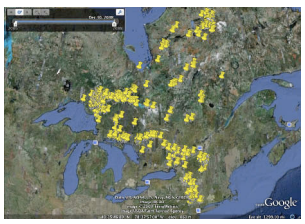


(b) Periodic behavior for $T = 168$.

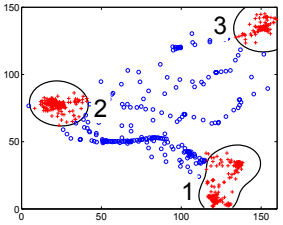
Figure 7: Periodic behaviors.

5.1.2 A bald eagle real case

We now test our method on a real dataset[2]. The data 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.

Figure 8: Real bald eagle data.

Figure 8(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 8(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 spots, 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 to-

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

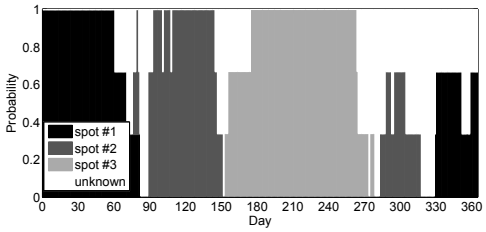


Figure 9: Periodic behaviors of bald eagle.

gether 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 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 9. 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."

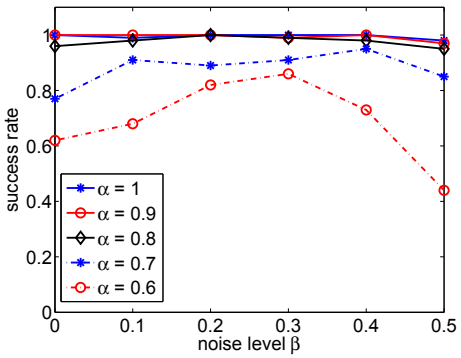
This real example shows the periodic behaviors mined from the movement provides an insightful explanation for the movement data.

5.2 Performance Evaluation

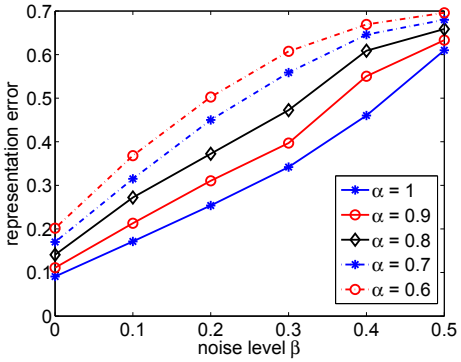
We further verify the effectiveness of our algorithms with respect to the two parameters we introduced at the beginning of this section, α and β, on synthetic datasets. Recall that α represents the proportion of regular segments in the whole sequence and β indicates the level of random noise. Again we use our *Running Example* to generate the synthetic data. This time, we vary α from 1 to 0.6, and simultaneously, we choose β from 0 to 0.5. We test the effectiveness of the period detection algorithm and the summarization algorithm separately. All experiments are repeated 100 times and the results are averaged.

For the period detection algorithm, we report the success rates in Figure 10(a). Since we know the ground truth ($T = 24$), we judge a trial is successful if among all detected periods, the one with the large correlation value is within the range $[23, 25]$. The result suggests that our period detection algorithm is nearly perfect in all cases with $\alpha \leq 0.8$. It is also noticeable that, compared to irregular segments, our algorithm is more robust to random noise, which may be caused by the failure of tracking devices or transmission networks during the data acquisition process. Furthermore, since irregular segments often reflects the changes of behaviors in the movement, the sensitivity to the irregular segments is also desirable for our algorithm which is designed for mining periodic behaviors.

For the summarization algorithm, we show in Figure 10(b) the representation error for $K = 10$ as defined in Section 4.3.



(a) Success rate of the period detection algorithm.



(b) Representation error of the summarization algorithm.

Figure 10: Performance evaluation.

To see the significance of the result, observe that, for example, with $\alpha = 0.9$ and $\beta = 0.1$, if we use 10 clusters to summarize all the daily segments of one year, the representation error is about 0.2. This means that we can obtain compact high-quality summarization even with moderate amount of irregularity and noise. This further shows that our algorithm is indeed able to filter out redundancy between the segments which are generated by periodic behaviors and therefore reveals the true behaviors.

6. RELATED WORK

A number of *periodic pattern mining* techniques have been proposed in data mining literature. Han *et al.* [8, 7] propose the algorithms for mining frequent partial periodic patterns. In this problem setting, each timestamp corresponds to a set of items. The goal is to find the patterns that appear at least *min_sup* times. Yang *et al.* [19, 20, 15, 21] propose a series of works dealing with variations of periodic pattern mining, such as asynchronous patterns [19], surprising periodic patterns [20], patterns with gap penalties [21], and higher level patterns [15]. In [22], it further addresses the gap requirement problem in biologic sequences. Different from previous works which focus on the categorical data, Mamoulis *et al.* [12] detects the periodic patterns for moving objects. However, all these works are based on the definition of *frequent* periodic pattern mining with a strict *min_sup* threshold. They tend to output a large set of patterns, most of which are slightly different. Besides, frequent

periodic patterns cannot capture the statistical information as the periodic behaviors. Similar to our definition of periodic behavior, Indyk *et al.* [9] studies the problem of of discovering the most representative trend that repeats itself every T timestamps. However, they can only discover one trend for a given period T and such trend covers the whole time span.

There are also works addressing the automatic period detection problem [9, 19, 11, 2, 3, 5, 6]. [11] and [19] have developed a similar linear distance-based algorithm for discovering the potential periods regarding the symbols of the time series. But this method misses some valid periods since it only considers the adjacent inter-arrivals. In [3], a data structure, the abbreviated list table (ALT) is proposed to compute the periods and the pattern. But such period is based on the threshold of *min_sup* which is not appropriate in our problem. Indyk *et al.* [9] develops an $O(n \log^2 n)$ time complexity algorithm using sketch approaches to find representative trend where n is the length of sequence. But only one period is detected in the whole sequence. Berberidis *et al.* [2] detects the period candidates for each symbol using autocorrelation. Improved from [2], Elfeky *et al.* [5] proposes a more efficient convolution method which considers multiple symbols together while detecting the period. However, as addressed in Section 3.2, both autocorrelation and convolution will detect a large set of period candidates, most of which are redundant. In [6], a method based on time warping is proposed, which is robust in the presence of shifting noise but is less efficient with time complexity $O(n^3)$. The only work that discusses the period detection for moving object is [1]. However, as we illustrated in Section 3.2, this method is easily affected by the spatial noise.

7. CONCLUSION AND FUTURE WORK

In this paper, we address an important and difficult problem: periodic behavior mining for moving objects. We propose a two-stage algorithm, Periodica. In the first stage, periods are detected through reference spots using Fourier transform and autocorrelation. In the second stage, periodic behaviors are statistically summarized using hierarchical clustering method. Empirically studies show that our method can deal with both noisy and complicated cases. A case study on a real data demonstrates the effectiveness of our method in practice.

While our approach fixes some reference spots using spatial information only, it is interesting to dynamically detect reference spots integrating with temporal information. This could give a more precise estimation on the reference locations. Another important issue is to find periodic behaviors in the data with the very sparse and inconstant sampling rate. We consider these as promising future works.

8. ACKNOWLEDGEMENT

The work was supported in part by the Boeing company, NSF BDI-07-Movebank, NSF CCF-0905014 (Cyber-Physical Systems), U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and by the U.S. Army Research Laboratory under Cooperative Agreement NumberW911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

The first author would like to thanks Xide Lin, Rui Li and Tianyi Wu for their valuable comments on this work.

9. REFERENCES

[1] S. Bar-David, 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.

[2] C. Berberidis, W. G. Aref, M. J. Atallah, I. P. Vlahavas, and A. K. Elmagarmid. Multiple and partial periodicity mining in time series databases. In *ECAI*, 2002.

[3] H. Cao, D. W. Cheung, and N. Mamoulis. Discovering partial periodic patterns in discrete data sequences. In *PAKDD*, 2004.

[4] H. Cao, N. Mamoulis, and D. W. Cheung. Discovery of periodic patterns in spatiotemporal sequences. *IEEE Trans. Knowl. Data Eng.*, 19(4), 2007.

[5] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid. Periodicity detection in time series databases. *IEEE Trans. Knowl. Data Eng.*, 17(7), 2005.

[6] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid. Warp: Time warping for periodicity detection. In *ICDM*, 2005.

[7] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In *ICDE*, 1999.

[8] J. Han, W. Gong, and Y. Yin. Mining segment-wise periodic patterns in time-related databases. In *KDD*, 1998.

[9] P. Indyk, N. Koudas, and S. Muthukrishnan. Identifying representative trends in massive time series data sets using sketches. In *VLDB*, 2000.

[10] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *ICDE*, 2008.

[11] S. Ma and J. L. Hellerstein. Mining partially periodic event patterns with unknown periods. In *ICDE*, 2001.

[12] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung. Mining, indexing, and querying historical spatiotemporal data. In *KDD*, 2004.

[13] M. Vlachos, P. S. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In *SDM*, 2005.

[14] C. Wang and S. Parthasarathy. Summarizing itemset patterns using probabilistic models. In *KDD*, 2006.

[15] W. Wang, J. Yang, and P. S. Yu. Meta-patterns: Revealing hidden periodic patterns. In *ICDM*, 2001.

[16] B. J. Worton. Kernel methods for estimating the utilization distribution in home-range studies. In *Ecology*, volume 70, 1989.

[17] Y. Xia, Y. Tu, M. Atallah, and S. Prabhakar. Reducing data redundancy in location-based services. In *GeoSensor*, 2006.

[18] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In *KDD*, 2005.

[19] J. Yang, W. Wang, and P. S. Yu. Mining asynchronous periodic patterns in time series data. In *KDD*, 2000.

[20] J. Yang, W. Wang, and P. S. Yu. Infominer: mining surprising periodic patterns. In *KDD*, 2001.

[21] J. Yang, W. Wang, and P. S. Yu. Infominer+: Mining partial periodic patterns with gap penalties. In *ICDM*, 2002.

[22] M. Zhang, B. Kao, D. W.-L. Cheung, and K. Y. Yip. Mining periodic patterns with gap requirement from sequences. In *SIGMOD Conference*, 2005.