Pattern Mining from Trajectory GPS Data

Xiaoliang Geng, Hiroki Arimura
Graduate School of Information Sciece and Technology,
IST, Hokkaido University,
N14, W9, Sapporo 060-0814, Japan
Email: {gengxiaoliang,arim}@ist.hokudai.ac.jp

Takeaki Uno
National Institute of Informatics,
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
Email: uno@nii.jp

***Abstract*—In this paper, we consider data mining from large discrete trajectory data. We study closed pattern mining for the class of trajectory envelope patterns. First, we introduce the basic definition of trajectory data. Then, we present a depth-first search algorithm that finds all trajectory envelope patterns in a given database that satisfies constrants on maximum width, minimum length, and minimum frequency. Finally, we ran experiments on a real trajectory dataset to evaluate our algorithm.**

***Keywords*-trajectory mining, spatio-temporal data, GPS-trajectory, closed pattern mining, computation geometry**

I. Introduction

By the rapid progress of mobile devices and sensors, trajectory data from GPS and mobile sensors have been popular. Since trajectories are sequences of real-valued locations with errors and missing values, mining of these trajectory is not a straightforward task. Hence, research of trajectory mining has attracted a great deal of attention for recent years [1], [2].

A. Problem we consider

In this paper, we study a trajectory mining problem in a trajectory database for the class $\mathcal{EVP}$ of envelope patterns defined as follows. Let m and $n \geq 0$ be any nonnegative integers. Let $\mathcal{S} = \{ s_i \,|\, i = 1, \ldots, m \}$ be a collection of m two-dimensional discrete trajectories, called a *trajectory database*, whose trajectory is a sequence of 2-dim points $s_i = (p_j^i)_{j=1}^n$ in $\mathbb{R}^2$ having the same length n. Then, an *envelope pattern* of width $\theta > 0$ and length $\ell \geq 0$ is a triple $P = (E, s, t)$ of a sequence $E = (E_1, \ldots, E_\ell)$ of $\ell = t - s + 1$ rectangles in $\mathbb{R}^2$, and time steps $s \leq t$. P says that there are some set X of k moving objects that have locations close each other contained in squares $E_1, \ldots, E_\ell$ at consecutive ℓ time steps $\tau_s, \ldots, \tau_t$ in interval $I = [\tau_s..\tau_t]$. Fig. 1 shows an example of an envelope pattern.

B. Main results

In the above definition of $\mathcal{EVP}$, there can be potentially infinitely many similar patterns for the continuous nature of space domain. To overcome this problem, we then introduce the class $\mathcal{CEVP}$ of *closed envelope patterns*, and study a mining problem for closed and constrained envelope patterns. As a main result, we present an efficient algorithm **DFM** that, given maximum width θ, minimum length ℓ, and minimum frequency σ, finds all longest closed envelope patterns with width $\leq \theta$, length $\geq \ell$, and frequency $\geq \sigma$ in a trajectory database. The algorithm combines depth-first search, efficient closure computation, and hash-based duplication check to achieve complete mining of all closed envelope patterns.

Though the algorithm is complete to output all the solutions, unfortunately, the running time and space are exponential in m due to non-anti-monotone nature of constraints on σ and closure test. Thus, it does not have any proven output-sensitive complexity. To examine basic characteristics of our algorithm, we ran experiments on a real trajectory data.[1]

C. Related work

There are two lines of researches on trajectory mining: trajectory clustering [2] and disk-based trajectory pattern mining [3]. The most closely related work in the latter context is the study of flock pattern mining [3], [4], [5]. Laube *et al.* [3] introduced the class of *flock patterns*, which resemble to our envelope patterns except that each region is implicit and defined by a fixed-radius circle. Laube *et al.* [3] presented an $O(nm \log m + nmk^2)$ time and $O(nm)$ space algorithm for flock patterns of length one, and Gudmundsson *et al.* [4] presented an $(1 + \varepsilon)$-approximation algorithm. Benkert *et al.* [5] tackled the problem for patterns with general length, $\ell \geq 1$, and proposed an $(2+\varepsilon)$ approximation algorithm whose running time is polynomial in m and $\frac{1}{\varepsilon}$, but exponential in the length ℓ of a pattern. Our algorithm **DFM** has time complexity that is exponential in m.

D. Organization of this paper

Sec.II gives basic definitions, and Sec.III studies closed envelope patterns. Sec.IV presents our algorithm and Sec.V shows experimental results. Finally, Sec.VI concludes.

[1]GeoLife project, `http://research.microsoft.com/en-us/projects/geolife/`

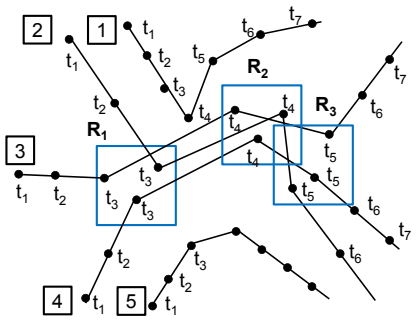

Figure 1. A 2-dim envelope pattern $P_1 = ((R_1, R_2, R_3), 3, 5)$ in $\mathbb{R}^2$ containing three moving objects $2, 3, 4$ at time steps t_3, t_4, t_5 in $I = [t_3..t_5]$ in a set of trajectories, where R_1, R_2, and R_3 are rectangles with width $\leq \theta$. This envelope pattern is not closed since the rectangular regions are not minimum bounding rectangles of their members.

II. Preliminaries

In this section, we give the basic definitions and concepts on trajectory pattern mining partly according to the Benkert *et al.* [5]. Although the following definitions can be easily generalized for dimension $d \geq 2$, we deal with only 2-dim case. For the definitions not found here, see textbooks of computational geometry (e.g., [6]).

A. Basic definitions

$\mathbb{N} = \{0, 1, 2, \ldots\}$ and $\mathbb{R}$ denote the sets of all natural and real numbers, respectively. For any $i, j \in \mathbb{N}$ $(i \leq j)$ and any $a, b \in \mathbb{R}$ $(a \leq b)$, we define the intervals $[i..j] = \{i, \ldots, j\}$. and $[a, b] = \{\, x \in \mathbb{R} \,|\, a \leq x \leq b \,\}$. For a set A, A^* denotes the set of all possibly empty sequence over A. For a sequence $s = \langle a_1, \ldots, a_n \rangle$ of n elements from A, we define $|s| = n$, $s[i] = a_i$, and $s[i..j] = a_i a_{i+1} \cdots a_j$. The *empty sequence* is denoted by $\langle\rangle$.

A *point* in $\mathbb{R}^2$ is a pair $p = (x, y) = (p.x, p.y) \in \mathbb{R}^2$. For a rectangle $R = [x_0, x_1] \times [y_0, y_1]$ in $\mathbb{R}^2$, the *width* of R is given by $width(R) = \max\{|x_1 - x_0|, |y_1 - y_0|\}$. For a set $S \subseteq \mathbb{R}^2$ of points, let S_x and S_y be the sets of all x- and y-coordinates of the points in S. The *minimum bounding rectangle* (MBR) containing S is given by $MBR(S) = [x_0, x_1] \times [y_0, y_1]$, where $z_0 = \min S_z$ and $z_1 = \max S_z$ for every $z \in \{x, y\}$. If $S = \emptyset$, then $MBR(S) = \emptyset$, too.

Lemma 1: Let S be any set of points. any rectangle R in $\mathbb{R}^2$, (i) $MBR(R \cap S) \subseteq R$, (ii) $X \subseteq Y$ implies $MBR(X) \subseteq MBR(Y)$, and (iii) $MBR(\widehat{R} \cap S) = \widehat{R}$ for $\widehat{R} = MBR(R \cap S)$.

Let $(A, \leq)$ is a partially ordered set. The *greatest lower bound* (*GLB*) of a subset $X \subseteq A$ is the element $\hat{x} \in A$ that satisfies the following (i) and (ii): (i) $\hat{x} \leq y$ for any $y \in X$ (a lowerbound of X), and (ii) for any $x \in A$, if x is a lowerbound of X then $x \leq \hat{x}$ holds.

B. Two-dimensional discrete trajectory data

Let $\mathcal{O} = \{1, \ldots, m\}$ be a set of m moving objects, or their object IDs (OID, for short), and let $\tau_1, \ldots, \tau_m$ be specified time steps with $\tau_{j-1} < \tau_j$ for every $j > 1$. Let $\mathcal{S} = \{s_1, \ldots, s_m\}$ be a trajectory database, whose element s_i $(1 \leq i \leq m)$ is a sequence $s_i = (p_j^i)_{i=1}^n$ of n points in $\mathbb{R}^2$, called a *discrete trajectory*. Each point $p_j^i = (x_j^i, y_j^i) \in \mathbb{R}^2$ stands for the location of object $1 \leq i \leq m$ at time step $1 \leq j \leq n$. Note that all trajectories in $\mathcal{S}$ have exactly same length in our definition. Then, we define $|\mathcal{S}| = m$ and $||\mathcal{S}|| = \sum_{i=1}^m |s_i| = mn$.

In the case for either an unevenly sampled sequence $\hat{s}_i$, or a continuous function $\hat{s}_i : [0, \infty) \to \mathbb{R}^2$ as in real trajectory data, we simply transform it to an evenly sampled trajectory $s_i = (p_j^i)_{i=1}^n$ by sampling n consecutive points from $\hat{s}$ at the specified time steps $t_1, \ldots, t_n$. We also assume that $\mathcal{O}$, $\mathcal{S}$, m, and n are fixed, otherwise stated.

C. Trajectory envelope patterns

A *2-dim trajectory envelope pattern* (or an *envelope pattern*, EVP) of length ℓ in $\mathcal{S}$ is a triple $P = (E, s, t)$, where $E = \langle R_1, \ldots, R_\ell \rangle$ is a ℓ-tuple of 2-dim rectangular regions in $\mathbb{R}^2$, called an *envelope*, and s, t $(1 \leq s \leq t \leq n)$ are indices in $[1..n]$ such that $\ell = t - s + 1$, called the *start* and *end positions* of P, respectively. The *width* of P is $w = \max_{1 \leq j \leq k} width(R_j)$. We denote by $start(P) = s$, $end(P) = t$, $len(P) = k$, and $width(P) = w$. If $\ell = 0$, then $P_0 = (\langle\rangle, s, s - 1)$ is the *empty pattern*, and we define $len(P_0) = 0$ and $width(P_0) = 0$.

Example 1: In Fig. 1, we show an example of an envelope pattern $P_1 = ((R_1, R_2, R_3), 3, 5)$ of length 3. We see that P_1 contains three moving objects $2, 3$, and 4 among five objects at time steps t_3, t_4, and t_5 in interval $I = [t_3, t_5]$.

For any $1 \leq s \leq t \leq n$, we denote by $\mathcal{EVP}(s, t)$ the class of all envelope patterns $P = (E, s, t)$ with starting and ending positions s and t. Then, we define $\mathcal{EVP}(s) = \cup_{t: s \leq t} \mathcal{EVP}(s, t)$ and $\mathcal{EVP} = \cup_s \mathcal{EVP}(s)$. Obviously, the class $\mathcal{EVP}$ is uncountable since all rectangles in $\mathbb{R}^2$ is uncountable.

Next, we define the semantics of envelope patterns by their cover sets. For every $1 \leq i \leq m$, the i-th trajectory s_i is *contained in* an envelope pattern $P = (E, s, t)$ of length ℓ if for every $1 \leq j \leq \ell$, the $(s+j-1)$-th point is contained in the j-th rectangle R_j, i.e., $p_{s+j-1}^i \in R_j$. In other words, s_i is contained in P iff the subsequence $(p_s^i, \ldots, p_t^i)$ is contained in the set $R_1 \times \cdots \times R_\ell$.

Definition 1 (cover set): The *cover set* of an envelope pattern P in $\mathcal{S}$ is the OID set $\mathbf{Cov}_{\mathcal{S}}(P) \subseteq \mathcal{O}$ defined by: for every $i \in \mathcal{O}$, $i \in \mathbf{Cov}_{\mathcal{S}}(P)$ iff s_i is contained in P.

Finally, we define the *frequency* of P in $\mathcal{S}$ by $freq_{\mathcal{S}}(P) = |\mathbf{Cov}_{\mathcal{S}}(P)| \geq 0$, the number of all trajectories of $\mathcal{S}$ that P contains.

III. Closed Trajectory Envelope Patterns

A. Envelope patterns with fixed start and end time steps

Let $s \leq t$ be any time steps. We introduce a binary relation $\sqsubseteq_{s,t}$ on $\mathcal{EVP}(s,t)$, called the *specificity relation*, as follows. For any envelope pattern $P_i = (E_i, s, t) \in \mathcal{EVP}(s,t)$, where $i = 1, 2$, with specified start and end positions s and t, we say that E_1 is *more specific* to E_2, denoted by $E_1 \sqsubseteq_{s,t} E_2$, if $E_1[j] \subseteq E_2[j]$ for every $1 \leq j \leq \ell$, where $\ell = t - s + 1$. E_1 is *properly more specific* to E_2, denoted by $E_1 \sqsubset_{s,t} E_2$ if $E_1 \sqsubseteq_{s,t} E_2$ and $E_2 \not\sqsubseteq_{s,t} E_1$ hold.

Lemma 2: The relation $\sqsubseteq_{s,t}$ on $\mathcal{EVP}(s,t)$ is a partial order.

Proof: We can easily verify that the reflexivity, transitivity, and anti-symmetricity hold on $\mathcal{EVP}(s,t)$. ■

Lemma 3: Let $X \subseteq \mathcal{O}$ be any subset of OIDs. Then, there exists the greatest lower bound (GLB) of all envelope patterns P in $\mathcal{EVP}(s,t)$ whose cover contains X w.r.t. $\sqsubseteq_{s,t}$.

Proof: Let $\ell = t - s + 1$. Firstly, we can compute the GLB $\widehat{P} = (\widehat{E}, s, t)$ by the following procedure: If $\ell = 0$, then we put $\widehat{P} = (\langle\rangle, s, t)$. Otherwise, for each $j \leftarrow 1, \ldots, \ell$, let $S[j] = \{\, p_j^i \mid i \in X \,\}$ be a set of points at time step j, and then let $\widehat{E}[j] = MBR(S[j])$. Finally, we set $\widehat{E}$ to be the ℓ-tuple $\widehat{E} = \langle \widehat{E}[1], \ldots, \widehat{E}[\ell] \rangle$. Next, we show that $\widehat{P}$ is actually the GLB as follows. Let $P = (E, s, t)$ be any EVP whose cover contains X. Then, $X[j] \subseteq E[j]$ holds for every j. Since $\widehat{E}[j]$ is the MBR of $X[j]$, we have $\widehat{E}[j] \subseteq E[j]$. Hence, it immediately follows that $\widehat{P} \sqsubseteq_{s,t} P$. ■

In the proof of Lemma 3, we can show that the GLB $\widehat{P}$ can be computed in $O(k\ell)$ time, where $k = |X|$.

B. One-sided closed envelope patterns

From now on, we consider the envelope patterns with specified start time and bounded width. Let $s \in [1..n]$ and $\theta > 0$ be any positive numbers. We denote by $\mathcal{EVP}(s,\theta)$ the set of envelope patterns P such that $start(P) = s$ and $width(P) \leq \theta$. Clearly, $\mathcal{EVP}(s,\theta) \subseteq \mathcal{EVP}(s)$ holds.

We extend the specificity relation $\sqsubseteq_{s,t}$ on $\mathcal{EVP}(s,t)$ to the binary relation $\sqsubseteq_s$, called the *one-sided specificity relation*, on the class $\mathcal{EVP}(s)$ of envelope patterns with same start time as follows. For any EVPs $P_i = (E_i, s, t_i) \in \mathcal{EVP}(s)$, we define $E_1 \sqsubseteq_s E_2$ if (i) $len(E_1) \geq len(E_2)$ holds (equivalently, $t_1 \geq t_2$), and (ii) $E[1..\ell] \sqsubseteq_{s,t} E_2[1..\ell]$ holds, where $t = t_2$ and $\ell = len(E_2) = t_2 - s + 1$. We also define $E_1 \sqsubset_s E_2$ if $E_1 \sqsubseteq_s E_2$ and $E_2 \not\sqsubseteq_s E_1$ hold.

Lemma 4: The relation $\sqsubseteq_s$ on $\mathcal{EVP}(s)$ is a partial order.

Proof: We can easily verify that the reflexivity, transitivity, and anti-symmetricity of $\sqsubseteq_s$ hold in $\mathcal{EVP}(s)$. ■

Below, we define a representative envelope pattern $\mathbf{CEVP}(X, s, \theta)$ in $\mathcal{EVP}(s,\theta)$ that is generated by a subset $X \subseteq \mathcal{O}$ of OIDs.

Definition 2: (longest width-bounded pattern generated by an OID set) Let $X \subseteq \mathcal{O}$ be any OID set, $s \in [1..n]$, and $\theta > 0$. We define the *envelope pattern* $\boldsymbol{CEVP}(X, s, \theta)$ *generated by* X, given s and θ, to be the unique pattern $\widehat{P} \in \mathcal{EVP}(s,\theta)$ that is the GLB of all envelope patterns in $\mathcal{EVP}(s,\theta)$ whose cover contains X w.r.t. $\sqsubseteq_s$. In other words, $\widehat{P}$ is a pattern that satisfies the following conditions 1) and 2):

1) $\widehat{P} \in \mathcal{EVP}(s,\theta)$ and $X \subseteq \mathbf{Cov}_{\mathcal{S}}(\widehat{P})$.
2) For any $P \in \mathcal{EVP}(s,\theta)$, if P satisfies $X \subseteq \mathbf{Cov}_{\mathcal{S}}(P)$, then $\widehat{P} \sqsubseteq_s P$ holds.

Algorithm 1 An algorithm that computes the one-sided closed envelope pattern in $\mathcal{EVP}(s,\theta)$ generated by an OID set X with start time s and maximum width θ.

```
1: procedure GetLongestEVP(X, s, θ)
2:     j ← 1; h ← s;
3:     while true do
4:         X[h] = { p_h^i | i ∈ X };
5:         Ê[j] ← the MBR of the set X[h];
6:         if width(Ê[j]) > θ then
7:             k ← j − 1; t ← h − 1; break;
8:         end if
9:         j ← j + 1; h ← h + 1;
10:    end while
11:    return P̂ = (⟨Ê[1], ..., Ê[k]⟩, s, t);
12: end procedure
```

Theorem 1 (existence and uniqueness of closed pattern): For any OID set $X \subseteq \mathcal{O}$, $s \in [1..n]$, and $\theta > 0$, the envelope pattern $\mathbf{CEVP}(X, s, \theta)$ always exists, unique, and can be computed in $O(\ell m)$ time.

Proof: We show that the procedure **GetLongestEVP** in Alg.1 computes $\mathbf{CEVP}(X, s, \theta)$. Suppose that $\widehat{P}$ is its output for X. Let $P \in \mathcal{EVP}(s,\theta)$ be any pattern of length $\ell \geq 0$ whose cover contains X. From Lemma 1, we can show that the iteration of the while-loop in Alg.1 never stop for any $1 \leq h \leq \ell$, and further $\widehat{P} \sqsubseteq_s P$ holds. Then, the uniqueness and time complexity immediately follows. ■

Now, we introduce our class of closed envelope patterns as follows.

Definition 3 (one-sided closed envelope pattern): A pattern $\widehat{P} \in \mathcal{EVP}(s,\theta)$ is a *(one-sided) closed envelope pattern* in $\mathcal{S}$ if $\widehat{P} = \mathbf{CEVP}(X, s, \theta)$ for some subset $X \subseteq \mathcal{O}$.

The set X is a generator of the closed pattern $\widehat{P}$, and called its *core set*. Note that a core set of $\widehat{P}$ is not unique. A closed pattern here is termed *one-sided* because it is obtained by expanding a pattern rightward as long as possible from fixed start point .

We denote by $\mathcal{CEVP}(s,\theta)$ the class of all closed envelope patterns within $\mathcal{EVP}(s,\theta)$ in $\mathcal{S}$. We also define $\mathcal{CEVP}(\theta) = \cup_s \mathcal{CEVP}(s,\theta)$ and $\mathcal{CEVP} = \cup_\theta \mathcal{CEVP}(\theta)$. We define the *closure* of $P \in \mathcal{EVP}(s,\theta)$ by the closed pattern $\mathbf{Clo}_{\mathcal{S}}(P) = \mathbf{CEVP}(\mathbf{Cov}_{\mathcal{S}}(P), start(P), \theta)$.

Lemma 5 (closure): The operator $\mathbf{Clo}_{\mathcal{S}}(\cdot)$ satisfies the following properties for any $P, Q \in \mathcal{EVP}(s,\theta)$:

1) $\mathbf{Clo}_{\mathcal{S}}(P) \sqsubseteq_s P$.
2) $P \sqsubseteq_s Q$ implies $\mathbf{Clo}_{\mathcal{S}}(P) \sqsubseteq_s \mathbf{Clo}_{\mathcal{S}}(Q)$.
3) $\mathbf{Clo}_{\mathcal{S}}(\mathbf{Clo}_{\mathcal{S}}(P)) = \mathbf{Clo}_{\mathcal{S}}(P)$.

Proof: By defining an operator $MB_S(R) = MBR(R \cap S)$ for rectangle R and any point set S, we can show the claim from Lemma 1. We omit the details. ■

Corollary 2 (characterization of closed envelope patterns): For any envelope pattern $P \in \mathcal{EVP}(s, \theta)$, the following conditions are equivalent each other:

1) P is a closed envelope pattern in $\mathcal{S}$
2) $P = \mathbf{CEVP}(X, s, \theta)$ for some $X \subseteq \mathcal{O}$.
3) $P = \mathbf{GetLongestEVP}(X, s, \theta)$ for some $X \subseteq \mathcal{O}$.
4) $\mathbf{Clo}_{\mathcal{S}}(P) = P$.

Proof: From Theorem 1 and definition of closed patterns, Conditions 1), 2), and 3) are equivalent each other. Then, it follows from definition of $\mathbf{Clo}_{\mathcal{S}}(\cdot)$ that 4) $\Rightarrow$ 2). Now, we show that 2) $\Rightarrow$ 4). Let $P = \mathbf{CEVP}(X, s, \theta)$ for some X (2). We can show that if $R = MBR(S)$ for some point set S, then $MBR(R \cap S) = R$. Applying this claim, we can that $\mathbf{Clo}_{\mathcal{S}}(P) = \mathbf{CEVP}(\mathbf{Cov}_{\mathcal{S}}(P), s, \theta)$ coincides to P. Thus, we have 2) $\Rightarrow$ 4). This completes the proof. ■

C. Closed constrained pattern mining problem

Even restricted to closed ones, the number of all possible EVPs in a given database $\mathcal{S}$ is prohibitive. Thus, we incorporate a set of constraints as well as closedness into our envelope pattern mining problem. Now, we state our problem as follows.

Closed Constrained Envelope Pattern Mining Problem:
Input: A collection $\mathcal{S} = \{s_1, \ldots, s_m\}$ of 2-dim trajectories with real numbers $\theta > 0$, $\ell > 0$, and an integer $\sigma \geq 1$.
Output: Find all closed envelope patterns P in $\mathcal{CEVP}$ appearing in $\mathcal{S}$ such that $width(P) \leq \theta$, $len(P) \geq \ell$, and $freq(P) \geq \sigma$ without duplicates.

The next theorem validates the use of closed patterns in our trajectory mining problem.

Theorem 3 (representative in constrained pattern mining): Let $s \in [1..n]$. If some pattern $P \in \mathcal{EVP}(s)$ satisfies the constraints on ℓ, θ, and σ, then there exists some closed pattern $\widehat{P} \in \mathcal{CEVP}(s)$ with $\mathbf{Cov}_{\mathcal{S}}(P) = \mathbf{Cov}_{\mathcal{S}}(\widehat{P})$ that satisfies the same constraints.

Proof: Let $P \in \mathcal{EVP}(s, \theta)$ be any pattern satisfying the constraints on ℓ, θ, and σ. Then, we can show that $\widehat{P} = \mathbf{Clo}_{\mathcal{S}}(P) \in \mathcal{CEVP}(s, \theta)$ satisfies the desired property that $\mathbf{Cov}_{\mathcal{S}}(P) = \mathbf{Cov}_{\mathcal{S}}(\widehat{P})$. Moreover, we can show that (i) $len(\widehat{P}) \geq len(P)$, (ii) $width(\widehat{P}) \leq width(P)$, (iii) $start(\widehat{P}) = start(P)$, and (iv) $freq(\widehat{P}) = freqt(P)$. Therefore, $\widehat{P}$ satisfies the constraints on ℓ, θ, and σ. ■

Theorem 3 above says that there is no loss of information even when we restrict our attention to closed envelope patterns in $\mathcal{CEVP}$. Our goal here is to devise a practically efficient algorithm that solves the above problem for large transaction databases.

Algorithm 2 The main algorithm for finding all closed envelope patterns with length $\geq \ell$, width $\leq \theta$, and frequency $\geq \sigma$ in a database $\mathcal{S} = \{s_1, \ldots, s_m\}$ of m trajectories of length n.

```
1: procedure DFM(S, θ, ℓ, σ)
2:     Θ ← (θ, ℓ, σ);
3:     HM ← ∅; //a hash table for envelope patterns
4:     for i ← 1, ..., m do
5:         for s ← 1, ..., n do
6:             RecDFM({i}, s, i, m, S, Θ, HM);
7:         end for
8:     end for
9: end procedure
```

IV. Algorithm

We present an algorithm **DFM** (Depth First Miner) for mining closed envelope patterns under given constraints from an input trajectory database.

A. Outline of our algorithm

In Alg.2 and Alg.3, we present the main algorithm **DFM** and its recursive subprocedure **RecDFM**, respectively. The main algorithm **DFM** in Alg.2 first receives an input trajectory database $\mathcal{S}$ and constraint parameters $\theta > 0$, $\ell > 0$, and $\sigma \geq 1$, and invokes the recursive subprocedure **RecDFM** in Alg.3 with each singleton OID set $X = \{i\}$, where $i \in \mathcal{O}$, and other arguments.

B. Recursive depth-first search algorithm *RecDFM*

Starting from a singleton OID set $X = \{i\}$, the subprocedure **RecDFM** in Alg.3 recursively generates all closed envelope patterns P satisfying given constraints by recursively expanding the current core set $X \subseteq \mathcal{O}$ using depth-first search approach such as Eclat algorithm. The algorithm **RecDFM** consists of the following steps.

1) Generation of core sets: At lines 1, 3, 4, and 13, the algorithm generates a child core set $Y \subseteq \mathcal{O}$ from the parent core set X using the depth-first search ([7]). In the DFS, the algorithm starts from a singleton set $X = \{i\}$ for each $i \in \mathcal{O}$. In an iteration, the algorithm expands X by adding a new object $i \in \mathcal{O}$ such that $i > k = \max(X)$. This ensures enumeration of core sets in a unique way. By incrementally maintaining $k = \max(X)$, this step can be implemented in $O(1)$ time.

2) Computation of the closed envelope pattern and its cover set: At line 5, the algorithm computes the longest and most specific envelope pattern $P = \mathbf{CEVP}(X, s, \theta)$ for a core set X generated using the subprocedure **GetLongestEVP**. At line 8, the algorithm computes the cover set $L = \mathbf{Cov}_{\mathcal{S}}(E)$ by simply scanning all of m trajectories one by one in $O(\ell)$ time per trajectory. These computations take $O(\ell m)$ time.

Algorithm 3 A DFS algorithm that recursively searches for closed and constrained envelope patterns with start time s in $\mathcal{CEVP}(s)$

```
1: procedure RecDFM(X, s, k, m, S, Θ, HM)
2:     (θ, ℓ, σ, Δ) ← Θ;
3:     for i ← k + 1, ..., m do
4:         Y ← X ∪ {i}; //Expanding a core set.
5:         P ← GetLongestEVP(Y, s, θ);
6:         if P.length < ℓ then
7:             continue; //Prune descendants by ℓ
8:         L ← Cov_S(P);
9:         if |L| ≥ σ and HM[(s, L)] = ⊥ then
10:            Output P as an answer;
11:            HM[(s, L] ← P;
12:        end if
13:        RecDFM(Y, s, i, m, S, Θ, HM);
14:    end for
15: end procedure
```

3) Checking anti-monotone constraint: At line 6, the algorithm checks if the obtained pattern P satisfies the minimum length constraint ℓ. The next lemma says that the pruning of the current core set X with θ and ℓ is sound and does not lose any successful branches.

Lemma 7: Let $s \in [1..n]$. Let $X_i \subseteq \mathcal{O}$ and $P_i = \textbf{CEVP}(X_i, s, \theta)$ for every $i = 1, 2$. Then, we have:

(a) $X_1 \subseteq X_2$ implies $width(P_1) \leq width(P_2)$.
(b) $X_1 \subseteq X_2$ implies $len(P_1) \geq len(P_2)$.

Proof: The lemma follows from the property that $X_1 \subseteq X_2$ implies $EVP_{\mathcal{S}}(X_1) \sqsubseteq EVP_{\mathcal{S}}(X_2)$. ■

4) Checking monotone constraint and duplicate detection: At line 9, the algorithm checks the frequency constraint σ of P and the duplicate detection for P. Since two distinct core sets X_1, X_2 can generate the same closed envelope pattern P, explicit test for duplicates is required by using a hash table HM that stores all discovered closed patterns P with the unique key $(start(P), \textbf{Cov}_{\mathcal{S}}(P))$ for P. Note that the above tests cannot prune the descendants since these constraints are not anti-monotone unlike those for θ and ℓ.

C. Analysis

Combining the above arguments, we give the correctness and the complexity of our main algorithm. Let M be the *output size*, i.e., the number of closed patterns in $\mathcal{S}$ as solutions and and N be the number of all core sets X that **RecDFM** examined. Clearly, $M \leq N \leq n2^m$.

Theorem 4 (main theorem): The algorithm **DFM** of Alg. 2 correctly finds all closed trajectory envelope patterns $P \in \mathcal{CEVP}$ in $\mathcal{S}$ such that $width(P) \leq \theta$, $len(P) \geq \ell$, and $freq(P) \geq \sigma$ without duplicates in $O(\ell m N)$ time and $O(\ell M)$ space.

Unfortunately, **DFM** is not a depth-first search algorithm in strict sense as in [7] since we cannot make complete pruning by length constraint and duplicate detection. To cope with this, the algorithm uses a hash table HM to avoid duplicates, but it still generates N candidate core sets. Due to this fact, the algorithm does not have output-polynomial time complexity. Furthermore, since the hash table stores M entries, it may require exponential space in the worst case.

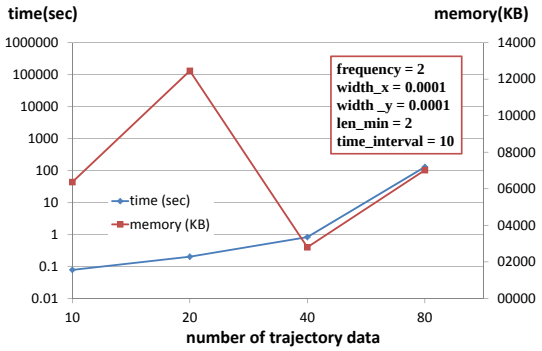

Figure 2. Running time and memory against input size

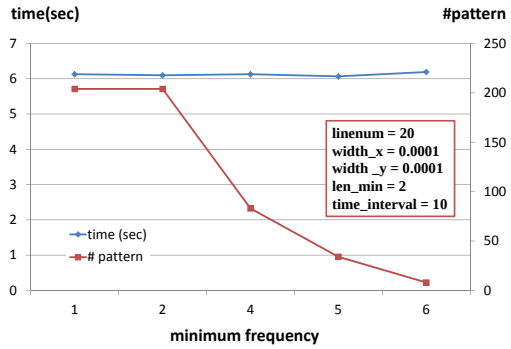

Figure 3. Running time against minimum frequency

V. Experiments

Finally, we report preliminary computational experiments to examine basic properties of the proposed algorithm **DFM** on real trajectory dataset.

A. Data and method

We used GPS trajectory dataset collected in Microsoft Research Asia, *GeoLife* project, consisting of 12,034 GPS-trajectories obtained from 165 pedestrians in 30 cities including Beijing from April 2007 to August 2009, containing over 18 million points in 795MB. As test data, we selected a small subset consisting of 80 trajectories with average length from 3 to 4 points, whose total size are 316 points in 14KB.

We implemented our algorithm **DFM** in C++ and compiled by g++ of GNU, version 4.5.3. We used a PC (Intel Core i7 CPU, 2.80GHz, 8GB of RAM) running Windows 7.

B. Results

In Fig. 2, we show the result on the running time (left) and memory usage (right) by varying the input size. We observed

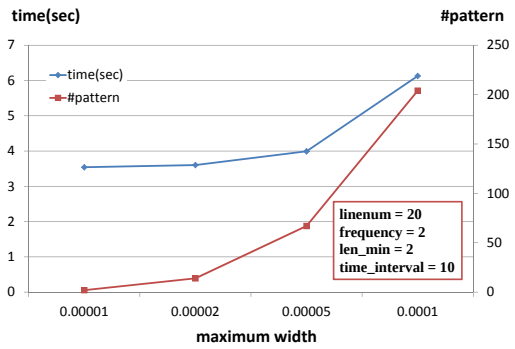

Figure 4. Running time against maximum width

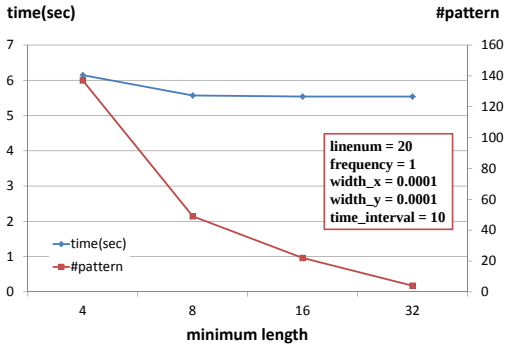

Figure 5. Running time against minimum length

that the running time increases as the input size increases, and we did not observe any clear dependency of memory usage on the input size. For example, the algorithm took 130 sec for finding six patterns in 80 trajectories. Hence, the current implementation is quite slow, and will take much time for large data sets.

In Fig. 3, Fig. 4, and Fig. 5, we show the results on the running time (left) and the number of solutions (right) by varying the minimum frequency σ, the maximum width θ, and the minimum length ℓ, respectively. In these figures, we first saw that the algorithm discovered from zero to six patterns in the data set depending on the values of parameters σ, θ, and ℓ as expected. In Fig. 4 and Fig. 5 with varying θ, and ℓ, we observed that the running time is almost proportional to the number of solutions. In Fig. 3 with varying σ, we did not observed such dependency.

Summary of Experiments: The above behavior of our algorithm was similar to other transaction data mining algorithms [7], [8]. By inspection, a large potion of running time was consumed in computing cover sets and duplicate detection, which is partly a reason of the large computation time of the current implementation of our algorithm **DFM**.

VI. Conclusion

In this paper, we studied the closed and constrained envelope pattern mining problem from trajectory data. We presented a depth-first mining algorithm **DFM** for the problem, and ran experiments on a real trajectory dataset to examine its basic behavior.

We list some of our future work. Though we assumed a database consisting of trajectories with same start time and the same length, it will be interesting to extend our framework for trajectories with various start times and lengths. From our experiments, we observed that a large potion of running time were spent for scanning redundant part of OID lists for a cover. Thus, it is important to skip redundant part by using geometric constraints [6]. Extension to two-sided closed patterns will be also interesting. Pruning in our DFS search was not complete due to monotonicity constraints such as minimum frequency and closedness. Therefore, it will be important to divise complete pruning methods such as PPC-extension of LCM [9].

Acknowledgment

The authors would like to thank Takuya Kida, Yuzuru Tanaka, Miki Haseyama, Shinichi Minato, Masayuki Takeda, Ayumi Shinohara, Yusaku Kaneta, Satoshi Yoshida, and Shuhei Denzumi for fruitful discussion on trajectory data mining. Finally, the authors thank anonymous reviewers for their comments which improved the correctness and the presentation of this paper very much.

References

[1] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. KDD'07*. ACM, 2007, pp. 330–339.

[2] J. Han, Z. Li, and L. A. Tang, "Mining moving object, trajectory and traffic data," in *Proc. DASFAA'10*, ser. LNCS, vol. 5982. Springer, 2010, pp. 485–486.

[3] P. Laube, M. van Kreveld, and S. Imfeld, "Finding REMO — detecting relative motion patterns in geospatial lifelines," in *Spatial Data Handling*. Springer, 2005, pp. 201–215.

[4] J. Gudmundsson, M. van Kreveld, and B. Speck, "Efficient detection of motion patterns in spatio-temporal sets," *GeoInformatica*, vol. 11, pp. 195–215, 2007.

[5] M. Benkert, J. Gudmundsson, F. Hubner, and T. Wolle, "Reporting flock patterns," *Computational Geometry*, vol. 41, pp. 111–125, 2008.

[6] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd ed. Springer-Verlag, 2000.

[7] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372–390, May/Jun 2000.

[8] J. Pei and J. Han, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. ICDE'01*. IEEE, 2001, pp. 215–224.

[9] T. Uno, T. Asai, Y. Uchida, and H. Arimura, "An efficient algorithm for enumerating closed patterns in transaction databases," in *Proc. 7th Int'l Conf. on Discovery Science (DS'04)*, ser. LNCS, vol. 3245. Springer, 2004, pp. 16–31.