



トップダウンアプローチによる大規模移動軌跡データからの頻出パターン発見

神野, 良太
上原, 邦昭

(Citation)

神戸大学都市安全研究センター研究報告, 15:201-211

(Issue Date)

2011-03

(Resource Type)

departmental bulletin paper

(Version)

Version of Record

(JaLCD0I)

<https://doi.org/10.24546/81011358>

(URL)

<https://hdl.handle.net/20.500.14094/81011358>



トップダウンアプローチによる大規模移動軌跡データからの頻出パターン発見

Trajectory Pattern Mining from Large Trajectory Data based on A Top-down Approach

神野良太¹⁾

Ryota Jinno

上原邦昭²⁾

Kuniaki Uehara

概要: 近年, GPS などの位置取得技術の普及により, 人や車の移動軌跡データが容易に入手できるようになった. これらのデータを大量に集めて解析し, 共通した行動を発見する Trajectory Pattern Mining と呼ばれる研究が注目されている. Trajectory Pattern Mining で発見される移動軌跡パターンは, 安全な道路, 交通サービスの考案にも有用とされるために, より複雑で意味のある移動軌跡パターンを検出することが求められている. しかしながら, 従来のグリッド領域ベースによる Trajectory Pattern Mining では, 空間的に大きいパターンと精度の細かいパターンを検出することがトレードオフの関係にあり, 複雑な移動軌跡パターンの発見は難しいとされている. 本研究では, 空間的に大きくかつ精度の細かいパターンの検出を可能にするために, 大きなパターンから細かいパターンと段階的に検出していく, トップダウンアプローチによる Trajectory Pattern Mining を提案する.

1. 序論

近年, GPS などの位置取得技術の普及により, 人や車の移動軌跡データを大量に集めて解析し, 共通した行動を発見する Trajectory Pattern Mining が注目されている. たとえば Giannotti らの研究 [1] では, 移動物体の移動時間と位置に関する共通したパターンの発見を行っている. 具体的には, まず2位置間の移動パターンを発見し, 短い移動パターンを結合して, より長い移動パターンを生成していく, ボトムアップの手法を用いている. このため, 複雑なパターンの発見が可能となっている. これらの既存研究で共通していることは, 空間情報をそのまま座標で扱うと計算コストが非常に大きくなるため, 空間を領域で区分し, 軌跡データの座標情報を該当する領域に置き換える手法を用いていることである. しかしながら, 適切な領域を定義することは非常に難しく, 空間の区分が細かすぎるとパターン自体の検出が難しくなり, 粗すぎると単純なパターンは検出できるが, 複雑なパターンの検出ができなくなる. このため, 適切な領域を定義することは大きな課題となっている.

本稿では, 複雑なパターンの検出を可能にするために, トップダウンアプローチによる Trajectory Pattern Mining を提案する. 本手法では, 粗い空間区分によって検出された単純なパターンからより小さい領域へと, 徐々に分割しながらパターンを検出し, 発見されたパターンを組み合わせで, 最終的に複雑なパターンを抽出するようにしている. このため, 空間区分を粗い領域区分から細かい空間区分まで階層性を持た

せた木構造で表現している．同様に，移動軌跡データや移動軌跡パターンも大きく単純な動きから小さく細かい動きまで階層性を持たせた木構造で表現している．また，空間区分には 4 分木のデータ構造を持たせており，分割された各領域には Z-ordering 法 [2] に基づいて番号を割り当てているため，異なる階層性を持つ領域，移動軌跡にも素早くアクセスできるようにしている．以下では，2 章で既存手法とその問題点について述べ，3 章で提案手法を説明する．4 章では平成 10 年東京都市圏パーソントリップデータセットに本手法を適用した実験と，その結果について述べる．最後に，5 章で本稿のまとめを述べる．

2. 既存手法

2.1 対象データ

Trajectory Pattern Mining は，移動物体の軌跡データから位置情報と移動時間に関して頻出するパターンを検出する手法である．対象となる物体の軌跡データは，位置情報とその時点の時間情報を一つの点とした連続データであり，次のような定義に従っている．

$$T = \langle (x_0, y_0, t_0), \dots, (x_k, y_k, t_k) \rangle$$

ここで $t_i (i = 0, \dots, n)$ は i 時点における時間， (x_i, y_i) は i 時点における位置を示している．通常，Trajectory Pattern Mining では，空間をいくつかの領域に分割し，2 次元で表現される位置情報 (x_i, y_i) を， (x_i, y_i) が含まれる領域 R_i に変換することで，位置情報を 1 次元情報として扱っている．たとえば図 2.1.1 では，空間をグリッド上に分割することにより，通過した領域とそのときの時間によって軌跡データの表現を可能としている．したがって，空間の領域定義によって軌跡データは次のように記述される．

$$T = \langle (R_0, t_0), \dots, (R_k, t_k) \rangle$$

このような表現により，データ処理自体が簡単になるだけでなく，パターン検出時の近傍を考慮する必要がなくなるために，検出アルゴリズムの開発が容易になるという利点がある．



図 2.1.1 位置情報の領域表現

2.2 T-Pattern

Trajectory Pattern Mining は，前節で説明したデータの集合から，物体の移動の位置情報，移動時間を考慮して移動パターン T-Pattern を検出する手法である．T-Pattern の定義は次のとおりである．

$$(S, A) = R_0 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_k} R_k$$

$$S = \langle R_0, \dots, R_k \rangle$$

$$A = \langle \alpha_1, \dots, \alpha_k \rangle$$

ここで $R_{i-1} \xrightarrow{\alpha_i} R_i$ は領域 R_{i-1} から領域 R_i に時間 α_i をかけて移動することを示している．すなわち，T-Pattern は i 時点における位置情報の連続データ S と， $i-1$ 時点から i 時点への移動時間 α_i の連続データ A で表現される．また T-Pattern (S, A) に対して，ある軌跡データ G が $(S, A) = R_0 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_k} R_k$ と近い移動をしているとき， G は T-Pattern (S, A) を支持しているという．たとえば図 2.2.1 の例を考える．パターン T が a から b へ時間 4 で移動しているのに対し，データ I は a から b へ時間 3 で移動しているため，移動時間の誤差は 1 である．また，パターン T が b から c へ時間 9 で移動しているのに対し，データ I は b から c へ時間 11 で移動しているため，移動時間の誤差は 2 である．つまり，移動時間の誤差を 2 以上許したときには，データ I はパターン T を支持していることになる．このように，移動時間の誤差を考慮すれば，個々の移動物体の軌跡データが異なっても，移動速度の違いに対応できるようになる．

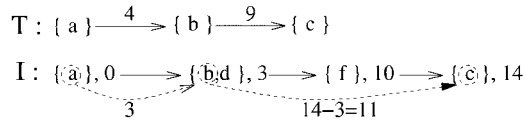


図 2.2.1 パターン支持の例

2.3 パターン検出

パターンを支持しているデータ数をそのパターンの支持度といい，支持度が大きいパターンほど多くのデータに共通した動きであるといえる．このため，最低支持度 σ を閾値として T-Pattern を検出している．この結果，最低支持度を満たさない，すなわち有用性の低いパターンの検出を防ぐようにしている．また，支持度は移動時間の誤差をどれだけ許すかによって変化するため，許容移動時間誤差 τ も決める必要がある．最終的に，Trajectory Pattern Mining は，許容移動時間誤差 τ のもとで最低支持度 σ を満たす T-Pattern を検出する問題となる．一方，ある T-Pattern は必ずその部分 T-Pattern よりも支持度が小さくなる．この特性を利用すると，最低支持度を満たす部分 T-Pattern のみを結合することによって，無駄な計算を避けて効率よく T-Pattern を検出することができる．このため，まず通過頻度の高い領域 Frequent Point を検出し，徐々に移動距離の長い T-Pattern の検出を行うようにしている．以下に，より具体的な検出手順を示す．

1. Frequent Point の検出
 - (a) σ 以上のデータが通過する領域を Frequent Point とする
2. 長さ 2 の T-Pattern の検出
 - (a) Frequent Point r_{prefix} , r_{suffix} を組み合わせる
 - (b) T-Pattern $r_{prefix} \xrightarrow{t_1} r_{suffix}$ の支持度が σ 以上になるような移動時間 t_1 を求める
 - (c) t_1 が存在すれば T-Pattern $r_{prefix} \xrightarrow{t_1} r_{suffix}$ を検出
3. 長さ $n-1$ の T-Pattern を結合して長さ n の T-Pattern を生成
 - (a) 長さ $n-1$ の T-Pattern TP_{prefix} , TP_{suffix} を選ぶ
 - (b) 次に示すように長さ $n-2$ の部分 T-Pattern が等しければ結合可能

$$TP_{prefix} = R_0 \xrightarrow{t_1} \dots \xrightarrow{t_{n-1}} R_{n-1}$$

$$TP_{suffix} = R_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} R_n$$

(c) 結合後の支持度が σ 以上であれば TP_{joined} を検出

$$TP_{joined} = R_0 \xrightarrow{t_1} \dots \xrightarrow{t_n} R_n$$

4. 最低支持度を満たすT-Patternが生成できなくなるまで3. を繰り返す

この結果、空間的に小さな移動軌跡パターンを基に、大規模なパターンの効率的な検出が可能となっている。しかしながら、ボトムアップアプローチの Trajectory Pattern Mining では必ずしも有用なパターンを検出することができないという問題がある。この問題点を次節で述べる。

2.4 問題定義

すでに述べたように、適切な領域を定義することは非常に難しい。空間の区分が細かすぎるとパターン自体の検出が難しくなり、粗すぎると単純なパターンは検出できるが、複雑なパターンの検出ができなくなる。このため、データに適切な領域を定義することは大きな課題となっている。たとえば、ボトムアップの手法では検出できない例を図 2.4.1, 2.4.2 に示す。図 2.4.1, 2.4.2 とともに、左図が 3 つの移動軌跡データ、中央図が領域変換された移動軌跡データ、右図がボトムアップで検出された移動パターン（実線矢印）を表している。図 2.4.1 の空間区分を細かくした場合、 $2 \rightarrow 1$, $14 \rightarrow 15$ のパターンは検出できるが、 $1 \rightarrow 14$ のパターンは検出できないため、結合によって $2 \rightarrow 15$ の大きなパターンが検出されることはない。また、図 2.4.2 の空間区分を粗くした場合、 $0 \rightarrow 1 \rightarrow 3$ の大きなパターンは検出できるが、3 つのデータに共通する移動パターンを忠実に表現することはできない。実際には、データや位置によって適切な空間区分の細かさが異なるため、空間区分の細かさは可変的に扱う必要がある。次章では、空間区分を階層的に扱い、空間的に大きく単純なパターンから徐々に部分的に空間区分を細かくしていくトップダウンアプローチによる Trajectory Pattern Mining について説明する。

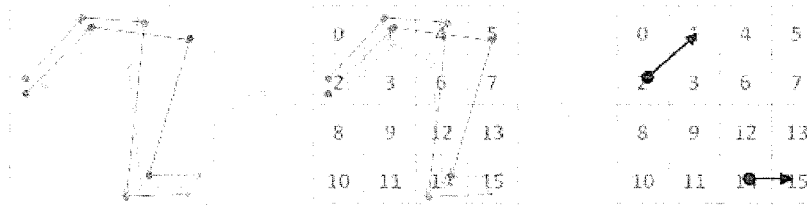


図 2.4.1 空間区分が細かい場合

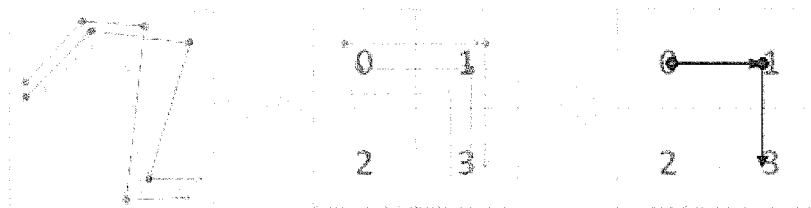


図 2.4.2 空間区分が粗い場合

3. トップダウンアプローチを用いたパターン検出

3.1 データ構造

既存手法と同様に、移動軌跡データの位置情報は領域表現しているものとする。本節では、一つのデータやパターンに対して複数の空間レベルを持たせた、木構造によるデータ構造を導入する。具体的には、空間レベルという領域の細かさの概念を領域表現に導入し、4 分木のデータ構造を用いる。また、各領域には Z-ordering 法に基づいて番号の割り当てを行う。Z-ordering 法による番号割り当てとは、図 3.1.1 のような、異なる空間レベルに対して階層性のある番号を割り当てる手法である。たとえば、図 3.1.2 のように領域番号を 2 進数で表現したとき、下位 2 ビットを除いた値は上位の空間レベルでの親領域を表しているため、親領域への素早いアクセスが可能となる。これらを組み合わせて、移動軌跡データ、移動軌跡パターンは、図 3.1.3 のような階層的な木構造で表現できる。ノードには自身が通過した領域情報が格納されているほか、その領域を 4 分割した、より細かい領域間での移動情報が子ノードのリストとして格納されている。このため、検出された移動軌跡パターンをより細かい精度で発見したいときには、パターンを支持するデータの木構造を掘下げ、さらに細かい空間レベルのデータを通過領域ごとに参照することができる。また、移動軌跡パターンに対しても木構造で表現しているため、ユーザの望む精度で検出されたパターンの情報参照が可能となる。

Grid Level 1		Grid Level 2				Grid Level 3															
0	1	0	1	4	5	0	1	4	5	16	17	20	21	2	3	6	7	18	19	22	23
		2	3	6	7	8	9	12	13	24	25	28	29	10	11	14	15	26	27	30	31
2	3	8	9	12	13	32	33	36	37	40	41	44	45	34	35	38	39	42	43	46	47
		10	11	14	15	40	41	44	45	48	49	52	53	42	43	46	47	50	51	54	55

図 3.1.1 Z-ordering 法による番号割り当て

Grid Level 1		Grid Level 2			
00	01	0000	0001	0100	0101
		0010	0011	0110	0111
10	11	1000	1001	1100	1101
		1010	1011	1110	1111

図 3.1.2 Z-ordering 法の階層性

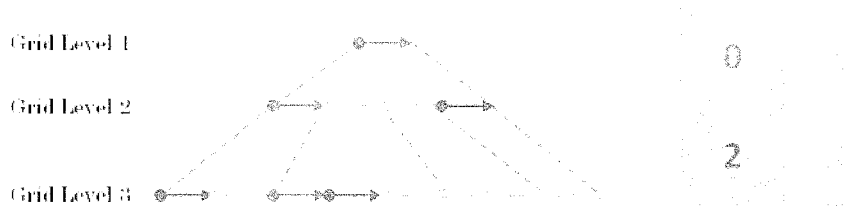


図 3.1.3 データ，パターンの木構造

3.2 トップダウンアプローチによるアルゴリズム

本節では，領域の粗い空間レベルで検出された，大きく単純なパターンに対し，パターンの通過した領域の中で，さらに細かい空間レベルのパターンの検出を行うというような，トップダウンアプローチによる Trajectory Pattern Mining を提案する．なお，大きなパターンを詳細化していく操作をドリルダウンと呼ぶ．本アルゴリズムを以下に示す．

1. 空間レベル 1 において 2 の手順を適用する
2. 空間レベル g におけるデータセット $D^{(g)}$ ，領域 $R^{(g)}$ からパターンを検出
 - (a) データ集合 $D^{(g)}$ から領域 R を通過するデータのみ取り出し，その集合を $D_R^{(g)}$ とする
 - (b) 領域 $R^{(g)}$ を 2×2 グリッドに展開することで，空間レベル $g + 1$ にする
 - (c) データ集合 $D_R^{(g+1)}$ に対して Trajectory Pattern Mining を適用する
3. 検出された T-Pattern $TP_i^{(g+1)}$ をドリルダウンする (検出された Frequent Point $R_i^{(g+1)}$ についても同様)
 - (a) データ集合 $D_R^{(g+1)}$ から T-Pattern $TP_i^{(g+1)}$ を支持するデータのみ取り出し，その集合を $D_{TP_i}^{(g+1)}$ とする
 - (b) T-Pattern $TP_i^{(g+1)}$ の通過する領域リスト $R_0^{(g+1)}, \dots, R_n^{(g+1)}$ とデータ集合 $D_{TP_i}^{(g+1)}$ に対して，再帰的に 2, 3 を繰り返す

上記の 2, 3 のステップが大きなパターンを詳細化していくドリルダウンにあたる．たとえば図 3.2.1 では，まず空間レベル 1 で 4 領域間での T-Pattern を考え， $0 \rightarrow 1 \rightarrow 3$ という T-Pattern が検出されている．さらに通過領域 0, 1, 3 を 4 分割して，空間レベル 2 でも 4 領域間のそれぞれで T-Pattern を考える．最後に，空間レベル 1 で検出された $0 \rightarrow 1 \rightarrow 3$ という T-Pattern，空間レベル 2 で検出された $2 \rightarrow 1, 14 \rightarrow 15$ という T-Pattern を統合させて， $2(2) \rightarrow 1(2) \rightarrow 1(1) \rightarrow 14(2) \rightarrow 15(2)$ という T-Pattern が完成されている．

4. 実験および結果

4.1 実験データ

実験データには，東京大学の空間情報科学研究センターから提供していただいた平成 10 年東京都市圏パーソントリップデータセットを用いている．実際に使用するデータは，データ検索サービスから得られる擬似的な緯度，経度，時間を連続データとして抽出したものである．また今回の実験では，東京都文京区を中心に $5 \sim 6\text{km}$ の範囲（正確には経度 $139.86546277617188 \sim 139.65946912382813$ ，緯度 $35.81054610640844 \sim 35.60522769359156$ の範囲）のデータから，100 人以上によって 500 回以上通過された Frequent Point を検

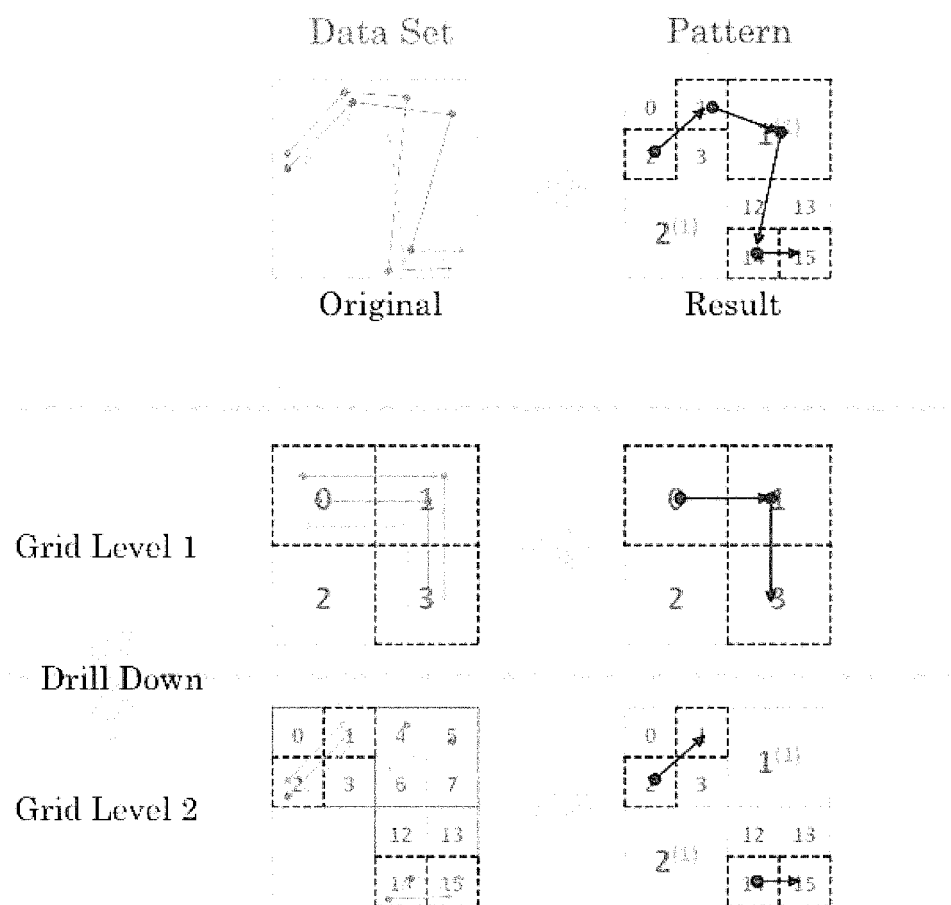


図 3.2.1 トップダウンアプローチによるパターン検出

出している。

4.2 実験結果

図 4.2.1 に空間レベル、範囲ごとの検出例を示す。空間レベルについては 1～4、範囲については縮尺 1, 2, 4 倍となっている。縮尺 2 倍については縮尺 1 倍範囲の破線正方形の範囲を拡大したもの、縮尺 4 倍については縮尺 2 倍範囲の破線正方形の範囲を拡大したものである。白塗りの正方形が検出された Frequent Point, すなわち 100 人以上によって 500 回以上通過された領域である。空間レベルが高くなるにつれ、Frequent Point が細かくなる一方、検出される範囲は少なくなっていくことが分かる。

図 4.2.2 は図 4.2.1 に示した縮尺 2 倍と 4 倍の範囲について、空間レベル 1～4 の Frequent Point を統合させた結果である。図 4.2.1 の縮尺 2 倍、4 倍のいずれにおいても、空間レベル 3 では領域区分が粗すぎて、どの部分を人が通過しやすいか分かりにくい。一方、空間レベル 4 では領域区分が細かすぎて Frequent Point が検出できていない部分が存在する。しかしながら、図 4.2.2 では、空間レベル 4 で検出できなかった広い範囲の Frequent Point を空間レベル 3 で検出された Frequent Point が補っており、一つの空間レベルでは表現できない複雑な頻度分布が表現できていることが分かる。

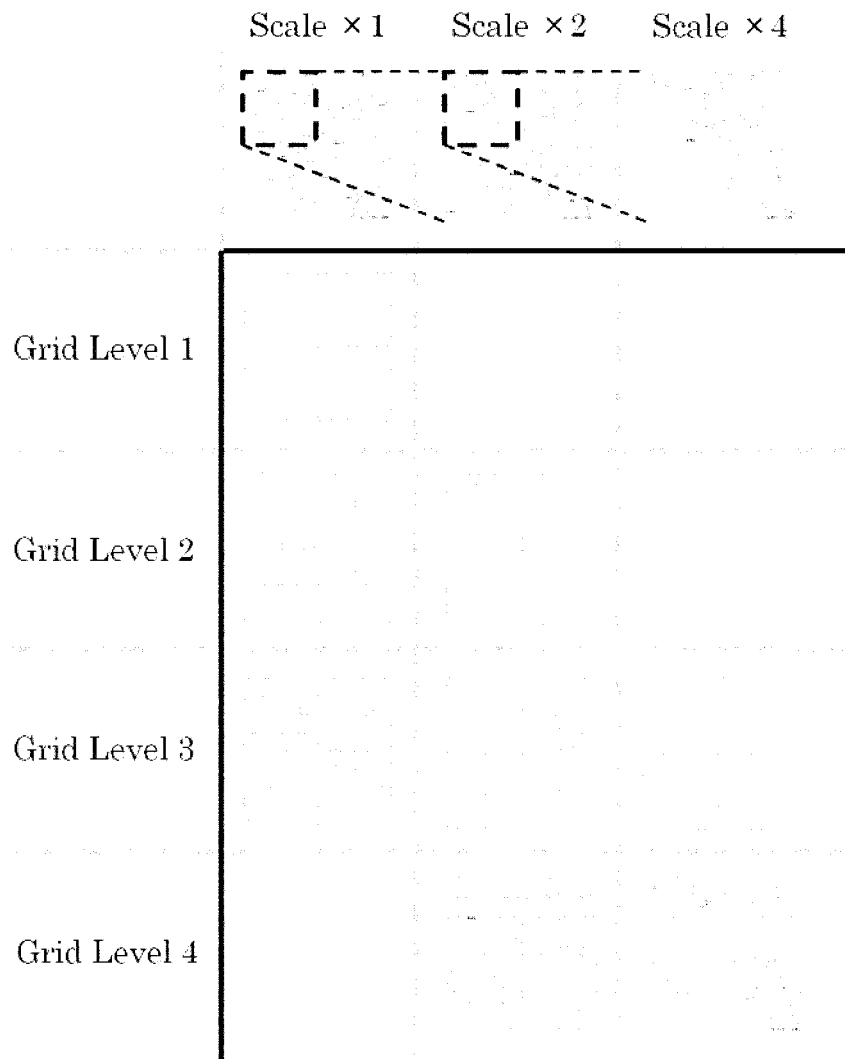


図 4.2.1 空間レベル，範囲毎に発見された Frequent Point

さらに，T-Pattern においても図 4.2.1 のように異なる空間レベルについて検出した上で，図 4.2.2 のように空間レベルで検出した T-Pattern を統合する必要がある．しかしながら，トップダウンアプローチによる Trajectory Pattern Mining は，現在，実装途中の段階であり，図 4.2.3 のように通過頻度の高い Frequent Point の検出部分に対してのみ実装している．今後の予定としては，トップダウンアプローチを適用して Frequent Point だけでなく T-Pattern の検出も可能にすることである．空間レベル 1 で検出された T-Pattern や Frequent Point をさらにドリルダウンして T-Pattern を検出するシステムを実装しなくてはならない．最終的に，図 4.2.4 のように異なる空間レベルの領域間の移動軌跡パターンを発見，表現ができるように実装を進めている．

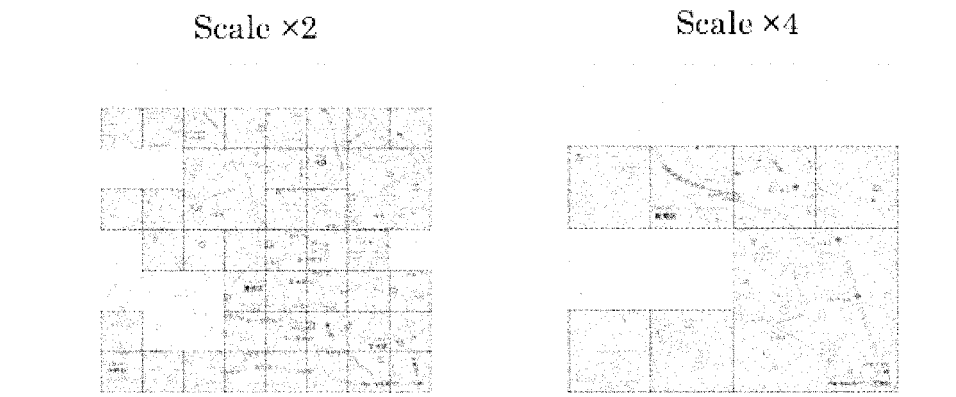


図 4.2.2 縮尺 2 倍（空間レベル 1～4），4 倍の範囲（空間レベル 1～4）における Frequent Point

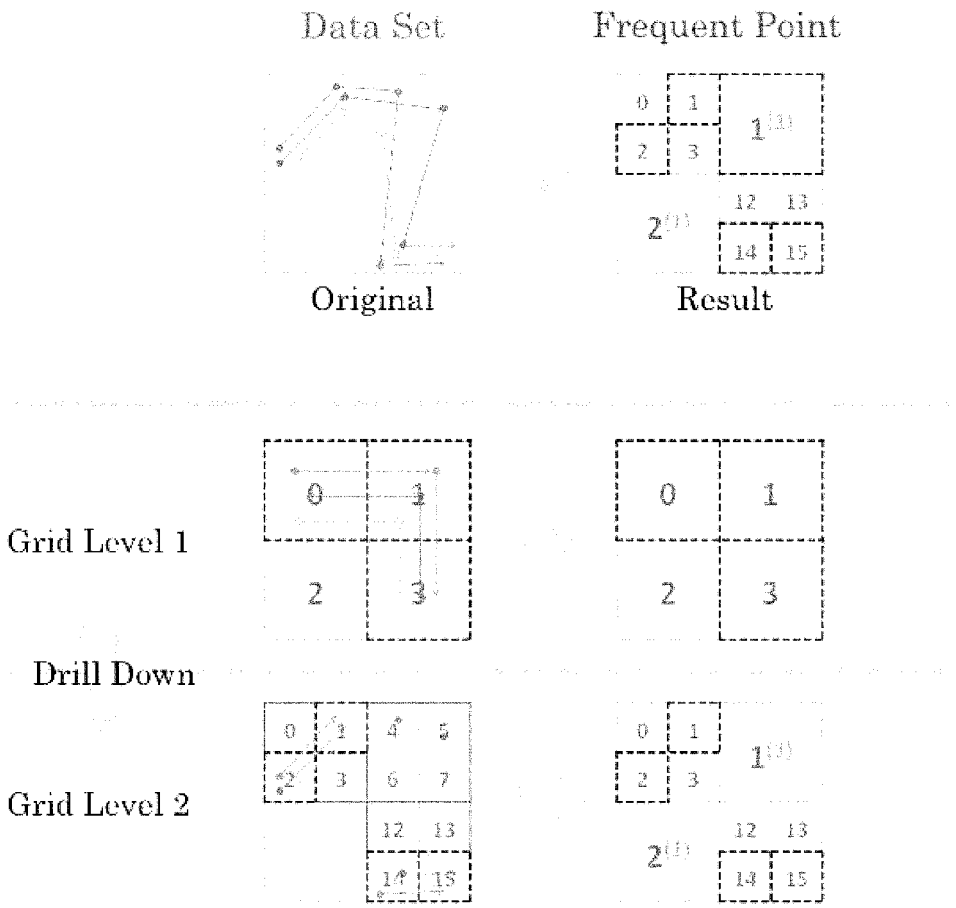


図 4.2.3 現時点での実装部分

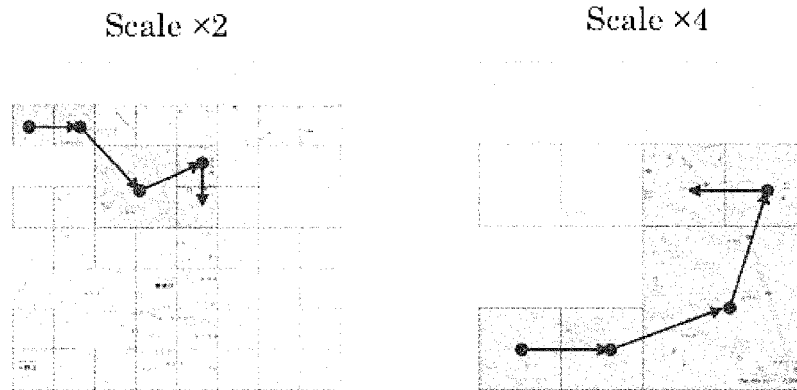


図 4.2.4 提案手法の検出例

5. まとめ

本稿では、領域ベースの移動軌跡パターン検出手法で検出の難しい、空間的に大きくかつ精度の細かいパターンを検出するために、空間区分に階層性を持たせたトップダウンアプローチによる Trajectory Pattern Mining を提案した。実験では、平成 10 年東京都市圏パーソントリップデータセットを用いて、文京区周辺における通過頻度の高い領域 Frequent Point を階層的に表現できることを示した。しかしながら、本研究はまだ実装段階であり、Frequent Point の検出においてのみトップダウンアプローチを適用している。今後、さらに Frequent Point を組み合わせた移動軌跡パターンの検出を実装する予定である。

6. 参考文献

- 1) Fosca Giannotti, Mirco Nanni, Dino Pedreschi and Fabio Pinelli: Trajectory Pattern Mining, Proc. of 13th KDD, pp.330-339 (2007).
- 2) Jack A. Orenstein: Spatial Query Processing in an Object-Oriented Database System, Proc. of SIGMOD '86, pp.326-336 (1986).
- 3) Florian Verhein: Mining Complex Spatio-Temporal Sequence Patterns. Proc. of SDM '09, pp.605-615 (2009).
- 4) Florian Verhein and Sanjay Chawla: Mining Spatio-Temporal Patterns in Object Mobility Databases. Data Mining and Knowledge Discovery, Vol.16, Issue 1, pp.5-38 (2007).
- 5) Jae-Gil Lee, Jiawei Han and Kyu-Young Whang: Trajectory Clustering: A Partition-and-Group Framework, Proc. of SIGMOD '07, pp.593-604 (2007).
- 6) Xiaoxi Du, Ruoming Jin, Liang Ding, Victor E. Lee and John H. Thornton Jr.: Migration Motif: A Spatial-Temporal Pattern Mining Approach for Financial Markets, Proc. of 15th KDD, pp.1135-1143 (2009).
- 7) Anna Monreale, Fabio Pinelli, Roberto Trasarti and Fosca Giannotti: WhereNext: A Location Predictor on Trajectory Pattern Mining, Proc. of 13th KDD, pp.637-646 (2007).

筆者：1) システム情報学研究科、学生、2) システム情報学研究科、教授

Trajectory Pattern Mining from Large Trajectory Data based on A Top-down Approach

Ryota Jinno

Kuniaki Uehara

Abstract

Trajectory is the path described by an object moving in space and time. Each data point in this path represents a certain position and time. Because of the increasing use of location-acquisition technologies, such as GPS for mobile devices, we can collect large spatio-temporal datasets, which can be used to discover useful knowledge about movement behavior from the datasets. This type of research is called “trajectory pattern mining.” This paper regards trajectory patterns as concise descriptions of frequent behaviors in terms of both space and time, following Giannotti’s work. We propose a new trajectory pattern mining algorithm based on a top-down approach.