

ストリームアルゴリズムを用いた移動軌跡パターンの検出

佐々木 健太, 上原 邦昭

神戸大学大学院 システム情報学研究科

連絡先: <sasaki@ai.cs.scitec.kobe-u.ac.jp>, <uehara@kobe-u.ac.jp>

(1) 動機: 近年, GPS などの位置情報技術の普及により, 人や車などの移動軌跡データを容易に入手できるようになった. これらの移動軌跡データを大量に集めて, 移動軌跡データマイニングを行い, 頻出する共通した行動を軌跡パターンとして検出する手法を提案する.

従来の移動軌跡データマイニング[1][2]は, ①全てのデータを一括で読み込み, 保持する, バッチ処理方式なので, メモリを大量に消費するという問題がある. 例えば図 1 の範囲で処理を行う場合, [1]のアプローチではメモリ空間が不足し, 計算すらできない. そこで, 並列分散処理できる MapReduce で問題を分割するか, マシンのスケールアップでメモリ空間を拡大するしか方法がなかった. さらに, ②全ての領域間に対して移動パターンを探索すると, パターンの検出数の増大に伴い, 計算時間も膨大になり, 人の流れデータの東京全体の解析すら扱えなくなるという問題もある. 例えば[2]の場合, 図1の範囲で軌跡パターンを検出するには, MapReduce でも計算に1時間以上かかってしまう. また, ③人の流れのように, 時間順でデータが到来し, 常に新しい情報が入力されるような, ストリームデータは分析することができない. さらに, ④ストリームデータは本来無限の入力となるため, データ全体を扱いきれないといった問題もあった. そこで, 本研究では, ストリームアルゴリズムにより移動軌跡データマイニングを拡張し, 軌跡パターンを検出することを検討する.

(2) アプローチ: ③④の問題に対して, ストリームアルゴリズムを採用する. ストリームアルゴリズムとは, 過去のデータの記憶に対して制限を設け, 過去の入力データ量に比べて非常に小さいメモリ空間のみを用いて計算を行う手法である. つまり, 巨大なデータがストリームとして入力されると, それをすべて記憶せずに, 必要な情報のみを抽出して蓄えるという技術である. また, ①②の問題に対しては, 人が多い領域に注目しながら, 必要なデータだけを保持していく方法をとる. 注目する領域を限定する手法は Trajectory Pattern Mining[1]でも採用されているが, 人口密度が一定に設定されている. そのため, 認識領域が広くなりすぎて解像度が低くなる場合がある. さらに[1]は, 注目領域同士をつなぐパターンの発見しか行っていないので, パターンの検出が粗くなる問題がある. しかし, 本研究では, パターン検出の解像度が低下することなく, 注目する領域を通るデータを詳細に分析して, 複雑なパターンを検出できるようなアプローチをとる. 具体的

には, ①の問題を解決するために, 注目領域の発見, 限定にも使用メモリ空間の削減を実現するアプローチをとる. すなわち, 注目領域すべてをメモリに展開するよりも, 小さい使用メモリ空間ですむように, ハッシュアルゴリズムを用いて実現する. ただし, この省メモリ手法では, ハッシュテーブルが小さい場合, 注目領域をキーとするハッシュテーブルでキーの衝突が起きることが多くなる. この問題は, 確率的な解析を行ったアプローチで近似解法により許容する. このようにすれば, 大規模なデータにも適用できる手法となる.

(3) 意義: 本研究では, 新しいデータ入力に常に対応できる. このため, 現在の生の移動軌跡データをモニタリングするだけでなく, 人が見つけることができない「現在」の軌跡パターンを発見できることを意味する. 現在の人の流れのパターンを解析すれば, 災害発生時に迅速なデータ解析による早急な対応への応用や, マーケティングにおけるイベントのターゲットを発見する事も考えられる. つまり, 情報伝達での迅速さを求める分野でも有用である.

(4) 参考文献:

[1] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, Fabio Pinelli. Trajectory Pattern Mining. Proc. of 13th KDD, pp.330-339, 2007.

[2] 神野, 熊南, 福井, 関, 上原. 階層グリッドを用いた四分木探索による移動軌跡データからの並列分散型頻出パターン検出. 人工知能学会論文誌, Vol. 27, No. 5, pp. 308-319, 2012.



図 1: 東京都の注目領域付近の軌跡パターン検出