DEIM Forum 2018 H7-5

MISCELA:センサ属性間の相関探索フレームワーク

原田　圭† 佐々木勇和† 鬼塚　真†

† 大阪大学大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1-5
E-mail: †{harada.kei,sasaki,onizuka}@ist.osaka-u.ac.jp

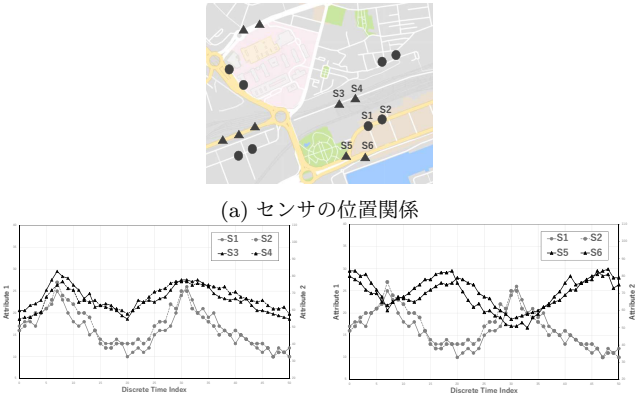
あらまし　近年，実世界の状態や現象を分析するため街中に環境センサを設置する都市が増加してきている．それに伴い，時空間データの共起を発見する共起探索問題の研究も盛んに行われている．多様な属性のセンサが混在している環境では，特に異なる属性同士で共起する組合せを発見することで，同じ属性のセンサ間の共起のみでは知り得ない多角的な情報を得ることができる．本稿では，属性間の共起を探索する問題を属性間の相関探索問題として新たに定義する．既存の共起探索手法を用いたアプローチは探索時に相関探索問題に不必要な計算を行うため非効率的である．そこで本稿では，効率的に属性間の相関を探索する相関探索フレームワーク MISCELA を提案する．実センサデータを用いた評価実験において，MISCELA は単純な相関探索手法と比較し探索時間を最大で 97% 短縮する．

キーワード　時空間データ，共起探索，多属性

1. はじめに

近年，情報技術や無線ネットワーク技術の高度化，センサデバイスの低価格化に伴い，街中に多数の環境センサを設置する都市が増加している．これらの都市では，環境センサを用いて収集した大量の時空間データを基に都市環境をモデル化し，都市計画の一部に IT システムを利用している．大量の時空間データから新たな知識や意味のある情報を抽出する技術は時空間データマイニングと呼ばれ，ビッグデータ分析技術の発展によって広く研究されている [1].

時空間データマイニング領域の研究課題の一つに空間的に近いセンサ間で時系列データが繰り返し同時に変化するパターンを発見する空間的な共起探索問題がある [2]．空間的に近い距離に位置するセンサ間で時系列データ上の共起パターンが観測された場合，それらのセンサは実世界で繰り返し発生する現象を協調的に観測している可能性が高い．共起パターンの発見は交通流分析による道路ネットワークマネジメントや都市の大気汚染分析などに役立つとされている．環境センサを導入している地域では多様な属性のセンサを設置していることが多い．気温や湿度，交通量といった多様な属性のセンサが混在する環境では，異なる属性のセンサ間においても共起を観測する可能性がある．特定の地域において属性間で同時変化するセンサの組合せを発見することはセンサデータを用いた都市環境マネジメントの高度化につながる．センサ属性間の同時変化は上昇と下降が同時に起こる変化と，上昇と下降が逆に起こる変化がある．特定の地域で属性間の同時変化を観測する場合，同時変化する属性間にはその地域特有の関係性が存在すると考えることができる．本稿では，地理的に近いセンサ集合内における二属性間の同時変化を属性間の相関と呼ぶ．図 1 は属性間の相関の例を示す．図 1(a) はセンサの設置位置を表し，円形，三角形の点はそれぞれ属性 a_1，a_2 をもつセンサを表す．また，センサ集合 $\{s_1, s_2\}$，$\{s_3, s_4\}$，$\{s_5, s_6\}$ はそれぞれ時系列データが同時上昇および同時下降する組合せである．図 1(b) に示すように $\{s_1, s_2\}$ の時系列データが上昇変化すると同時に $\{s_3, s_4\}$ の時系列データも上昇変化するため，二つの属性を含むセンサ集合 $\{s_1, s_2, s_3, s_4\}$ ではセンサ属性 a_1，a_2 間に正の相関がある．一方，図 1(c) に示すように $\{s_1, s_2\}$ の時系列データが上昇変化すると同時に $\{s_5, s_6\}$ の時系列データが下降変化するため，センサ集合 $\{s_1, s_2, s_5, s_6\}$ ではセンサ属性 a_1，a_2 間に負の相関がある．



(a) センサの位置関係

(b) 正の相関　(c) 負の相関

図 1　属性間の相関

本稿では，多種類の属性を含むセンサ集合から属性間の相関を発見する問題を新たに属性間の相関探索問題と定義する．属性間の相関探索問題では二属性間の同時変化を探索するために，変化率のしきい値を用いて各センサの時系列データ中から変化の大きい部分を抽出し，距離のしきい値を用いて地理的に近いセンサを選択する．これら二つのしきい値はユーザ定義であるが属性依存もしくはデータセット依存のパラメータであるため，適切な値を設定する作業はユーザにとって大きな負担となる．また，属性間の相関探索問題は共起探索問題の中で二属性間での同時変化の探索に特化した問題であり，多種類の属性を含むセンサ集合内から，二つの属性を含むセンサの組み合わせを探索する必要がある．既存の共起探索手法は探索にセンサ属性を

考慮しておらず，多種類の属性を含むセンサに対して適用した場合，二つの属性を含むセンサの組合せの他に単一の属性のみを含むセンサの組合せや三つ以上の属性を含むセンサの組合せの共起も探索するため，属性間の相関探索問題においては非効率的である．そこで本稿では，変化率のしきい値および距離のしきい値を手法中で自動決定するとともに，多種類の属性を含むセンサ集合から効率的に二つの属性を含むセンサの組合せを探索する属性間の相関探索フレームワーク MISCELA を提案する．MISCELA では変化率の大きい点が全点の $k\%$ になるように自動的にしきい値の決定を行う．また，OPTICS アルゴリズムを用いることでクラスタリングに必要な距離パラメータの決定を自動化する．加えて，多種類の属性を含むセンサ集合から二つの属性を含むセンサの組合せのみを探索することで，既存の共起探索手法を用いたアプローチと比較して属性間の相関探索を効率化する．本稿では気温，照度，騒音，交通量および湿度の 5 属性からなる実センサデータを用いて相関探索実験を行い，MISCELA は既存の共起探索手法を用いたアプローチと比較して探索時間を最大 97% 短縮する．

本稿の構成は次のとおりである．2. 章にて関連研究について説明し，3. 章で本稿の前提となる知識について述べる．4. 章にて本稿で新たに定義する属性間の相関探索問題を説明し，5. 章において提案手法について説明する．6. 章で提案手法の評価と分析を行い，7. 章にて本稿のまとめについて述べる．

2. 関連研究

本章では属性間の相関探索問題と関係のある区分線形表現，密度ベースクラスタリング，共起探索についての代表的な手法を説明する．

2.1 区分線形表現

属性間の相関探索問題は時間的に変化するデータを対象とするため，時系列データマイニング技術を利用することができる．時系列データマイニングの主な目的はデータの規則性や傾向，特徴を抽出することである．抽出した特性は異常値の検出，類似パターンのクラスタリング，将来予測などの様々な技術に応用することができる．時系列データマイニングではデータの表現方法が重要である．元データの情報を失うことなく時系列上の細かい変動やノイズの除去，データの圧縮等を行うために，Fourier 変換，Wavelet 変換，Symbolic マッピング，区分線系表現などのデータ表現技術が提案されている [3]．その中でも区分線形表現は時系列パターンのクラスタリングや時系列上の変化点検出などに適用しやすいことから広く利用されている．区分線系表現では時系列データを複数の線分セグメントによって近似することで，局所的な微小変動を取り除くことができデータの全域的な変化のみをとらえることが可能である．一般的に時系列データの区分線形化には sliding window，top-down，bottom-up といった手法がよく用いられる．

本稿で提案する相関探索手法では入力された時系列データを区分線形表現に変換して取り扱う．この際，その単純性から bottom-up アルゴリズムを採用する．bottom-up 手法では，はじめに隣接点同士を線分としてセグメント化する．その後，結合後のデータ誤差が最小となる隣接セグメントを繰り返し結合していく．結合するセグメントがなくなるか，最小データ誤差がしきい値 ε を超えた場合に結合処理を終了する．なお，本稿で提案する手法では bottom-up アルゴリズムの代わりにその他の区分線形化アルゴリズムを使用することも可能である．

2.2 密度ベースクラスタリング

属性間の相関探索問題では地理的に近いセンサ集合を求める必要がある．クラスタリング技術は最も基本的なデータマイニング技術の一つとして，空間データの分析に非常に頻繁に用いられている．基本的なクラスタリング手法の一つ密度ベースクラスタリングがある．一般的に，密度ベースクラスタリングには事前にクラスタ数を設定する必要がない，クラスタの形状が超球形に限定されないといった特徴がある [4]．密度ベースクラスタリングの代表的なアルゴリズムとしては，DBSCAN [5] や Mean Shift [6] が挙げられる．特に DBSCAN は密度ベースクラスタリングの基本アルゴリズムに位置付けられており広く研究されている．DBSCAN では有意なクラスタリング結果を得るためのパラメータ設定が困難であることが指摘されており，これを解決するために距離パラメータを変化させた場合のクラスタをまとめて抽出できる OPTICS [7] が提案された．

本稿ではデータセットとして街中に設置されているセンサ対象としており，それらはデータセットやセ属性ごとにセンサの設置間隔が異なる．そのため，MISCELA では距離パラメータの設定を最適化できる OPTICS アルゴリズムを用いてセンサのクラスタリングを行うことで空間的に近い距離にあるセンサ集合を求める．

2.3 共起探索

時系列データマイニングを利用した研究分野の一つに時系列データの共起探索問題がある．時系列データの共起とは複数の時系列間において測定値が繰り返し同時に変化するパターンのことを呼ぶ．時系列データの共起は同時上昇と同時下降だけには限らず上昇と下降の組合せも存在する．特に空間的に近い位置に設置されているセンサ群の中で確認される共起パターンを SCPs(Spatial Co-evolving Patterns) と呼ぶ．Zhang らは環境センサデータから SCPs を探索するアルゴリズム Assembler を提案した [2]．また [8] では，共起する時系列をクラスタリングし，動的な共起空間を発見する手法が提案されている．この手法は共起空間の推測や共起時系列の予測に応用することができる．

SCP 探索アルゴリズム Assembler は空間的に近いセンサの組合せを探索するために SCP 探索木と呼ばれる木構造を用いている．これは，センサ集合内から空間的に近いセンサの組合せを一度の深さ優先探索で探索できる構造である．本稿ではベースライン手法における空間的に近いセンサの探索手法にこの SCP 探索木を利用する．

3. 事前知識

本稿で定義する属性間の相関探索問題はセンサ間の共起探索問題を発展させた問題である．本章では相関探索問題および相関探索手法の事前知識を述べる．

地理空間情報を持った m 台のセンサの集合を $\boldsymbol{S} = \{s_1, s_2, \ldots, s_m\}$ とする．各センサ $s_i \in \boldsymbol{S}(1 \leqq i \leqq m)$ は位置 l_i に設置されており，属性 $a_i \in \boldsymbol{A}, \boldsymbol{A} = \{a_1, a_2, \ldots, a_p\}$ を持つ．属性は気温，騒音，照度といった観測値のカテゴリを表す．s_i は時間領域 $\boldsymbol{T} = \langle t_1, t_2, \ldots, t_n \rangle$ 上の各 t_j において a_i に対応する値を測定する．ここで $t_j (1 \leqq j \leqq n)$ はタイムスタンプを表し，これらは等間隔であるものとする．タイムスタンプ t_j における s_i の測定値を $s_i[t_j]$ と定義する．さらに測定値の変化率を式 (1) のように定義する．

$$r_i[t_j] = \frac{s_i[t_{j+1}] - s_i[t_j]}{t_{j+1} - t_j} \tag{1}$$

本研究では変化率の大きな点を下記のように定義する．

定義 1 (*evolving* タイムスタンプ) しきい値 $\Theta = (\theta^+, \theta^-)$ が与えられたとき，$r_i[t_j] \geqq \theta^+$ である場合，s_i は t_j で正の *evolving* であるといい，t_j を正の *evolving* タイムスタンプと呼ぶ．一方，$r_i[t_j] \leqq \theta^-$ である場合，s_i は t_j で負の *evolving* であるといい，t_j を負の *evolving* タイムスタンプと呼ぶ．また，しきい値 Θ を *evolving* 値と呼ぶ．

センサ集合 $\boldsymbol{S}$ が与えられたとき，地理的に近い関係にある $\boldsymbol{S}$ の部分集合および同じ属性をもつ $\boldsymbol{S}$ の部分集合を以下のように定義する．

定義 2 (近傍集合) しきい値 h と $\boldsymbol{S}$ の部分集合 $\boldsymbol{G} \subseteqq \boldsymbol{S}$ が与えられたとき，$\forall s \in \boldsymbol{G}, \exists s' \in \boldsymbol{G} - \{s\}\ s.t.\ dist(s, s') \leqq h$ であるならば $\boldsymbol{G}$ は地理的に近い関係にあるセンサ集合であると定義する．このとき $\boldsymbol{G}$ を近傍集合と呼ぶ．ここで $dist(s, s')$ は s と s' の空間的な距離を表す．また，しきい値 h を近傍半径と呼ぶ．

定義 3 (同属性集合) センサの属性 a と $\boldsymbol{S}$ の部分集合 $\boldsymbol{S}_a \subseteqq \boldsymbol{S}$ が与えられたとき，$\forall s \in \boldsymbol{S}_a$ が同じ属性 a をもつならば，$\boldsymbol{S}_a$ を a の同属性集合と呼ぶ．

定義 4 (近傍同属性集合) 属性 a の同属性集合 $\boldsymbol{S}_a$ の部分集合 $\boldsymbol{G}_a \in \boldsymbol{S}_a$ が与えられたとき，$\boldsymbol{G}_a$ が近傍集合であるならば，$\boldsymbol{G}_a$ を近傍同属性集合と呼ぶ．

センサ間および属性間の共起は下記のように定義する．

定義 5 (空間的な共起) 近傍集合を $\boldsymbol{G}$，$\boldsymbol{G}$ 内のセンサ $s_i \in \boldsymbol{G}$ が $t_j \in \boldsymbol{T}$ で *evolving* であるためのしきい値を Θ とする．あるタイムスタンプ t_j に対して，$\forall s_i \in \boldsymbol{G}, r_i[t_j] \geqq \theta^+$ であるとき t_j は Θ に正で共起するといい，$t_j \xrightarrow{+} \Theta$ と表す．また $\forall s_i \in \boldsymbol{G}, r_i[t_j] \leqq \theta^-$ であるとき t_j は Θ に負で共起するといい，$t_j \xrightarrow{-} \Theta$ と表す．正または負で共起するタイムスタンプの集合を $E(\boldsymbol{G}) = \{t_j \in \boldsymbol{T} | t_j \xrightarrow{\pm} \Theta \vee t_j \xrightarrow{-} \Theta\}$ と表し，これを $\boldsymbol{G}$ の空間的な共起と呼ぶ．

定義 6 (属性間の空間的な共起) 二つの近傍同属性集合 $\boldsymbol{G}_{a_1}, \boldsymbol{G}_{a_2} (a_1 \neq a_2)$ について $s_i \in \boldsymbol{G}_{a_1}$ が $t_j \in \boldsymbol{T}$ で *evolving* であるためのしきい値を Θ_{a_1}，$s_i \in \boldsymbol{G}_{a_2}$ が $t_j \in \boldsymbol{T}$ で *evolving* であるためのしきい値を Θ_{a_2} とする．$\boldsymbol{G}_{a_1} \cup \boldsymbol{G}_{a_2}$ が近傍集合であるとき，$P_{a_1,a_2} = \{t_j \in E(\boldsymbol{G}_{a_1}) \cap E(\boldsymbol{G}_{a_2}) | (t_j \xrightarrow{+} \Theta_{a_1} \wedge t_j \xrightarrow{+} \Theta_{a_2}) \vee (t_j \xrightarrow{-} \Theta_{a_1} \wedge t_j \xrightarrow{-} \Theta_{a_2})\}$ を $\boldsymbol{G}_{a_1}$ と $\boldsymbol{G}_{a_2}$ 間の正の空間的な共起と呼ぶ．また $N_{a_1,a_2} = \{t_j \in E(\boldsymbol{G}_{a_1}) \cap E(\boldsymbol{G}_{a_2}) | (t_j \xrightarrow{+} \Theta_{a_1} \wedge t_j \xrightarrow{-} \Theta_{a_2}) \vee (t_j \xrightarrow{-} \Theta_{a_1} \wedge t_j \xrightarrow{+} \Theta_{a_2})\}$ を $\boldsymbol{G}_{a_1}$ と $\boldsymbol{G}_{a_2}$ 間の負の空間的な共起と呼ぶ．

4. 問題定義

本章では本稿で新たに定義する属性間の相関探索問題について説明する．

定義 7 (属性間の相関探索問題) 二つの近傍同属性集合 $\boldsymbol{G}_{a_1}, \boldsymbol{G}_{a_2} (a_1 \neq a_2)$ 間の正の空間的な共起を P_{a_1,a_2}，負の空間的な共起を N_{a_1,a_2} とする．しきい値 ψ が与えられたとき，$|P_{a_1,a_2}| \geqq \psi$ であるならば，属性 a_1, a_2 間にセンサ群 $\boldsymbol{G}_{a_1}, \boldsymbol{G}_{a_2}$ 上で正の相関があるという．一方，$|N_{a_1,a_2}| \geqq \psi$ であるならば，属性 a_1, a_2 間にセンサ群 $\boldsymbol{G}_{a_1}, \boldsymbol{G}_{a_2}$ 上で負の相関があるという．また，しきい値 ψ を最小サポートと呼ぶ．属性間の相関探索問題では，正または負の相関があるすべての近傍同属性集合を列挙する．

実世界で発生する現象はその発生原因として多くの要素を持つと考えられており，都市環境センシングでは，収集した大量のセンサデータを基に都市で発生している現象の原因をできるだけ詳しく分析することが重要である．属性間の相関を発見することで，実世界で発生した同じ現象によって影響を受けているセンサ属性を発見することができ，単一属性のセンサ間における共起の発見のみからでは知り得ない多角的な情報を抽出することができる．

本稿では，属性間の相関探索問題に対し，四つのステップで属性間の相関探索を実施する手法をベースライン手法とする．まず，入力された各センサの時系列データに対して区分線形化アルゴリズムを適用し，入力された時系列データを線分セグメントで近似する．ここでは単純性を考慮し，広く利用されている bottom-up 手法 [3] を用いる．次に，各時系列データ中の *evolving* タイムスタンプを抽出する．時系列データ中の各タイムスタンプの変化率がしきい値 θ^+ 以上であるとき，そのタイムスタンプを正の *evoloving* タイムスタンプと判定する．また θ^- 以下であるとき，そのタイムスタンプを負の *evolving* タイムスタンプと判定する．なお，変化率のしきい値 θ^+ および θ^- は入力パラメータとして属性ごとにそれぞれユーザ定義で与える．続いて，空間的に近いセンサを求めるためにセンサ集合をクラスタリングする．ベースライン手法ではクラスタリングアルゴリズムとして DBSCAN [5] を用いる．なお，DBSCAN の入力パラメータである距離のしきい値 h はユーザ定義で与えるとし，$MinPts$ は 2 とする．最後に，求めた各クラスタに対して相関探索を行い属性間の相関を出力する．相関探索は SCP 探索木 [2] を利用して行う．SCP 探索木は与えられた一つの連結グラフに対応して構築される木構造である．本稿では連結グ

ラフを以下のように定義する．

定義 8 (連結グラフ) センサ集合 $\boldsymbol{S}$ と近傍半径 h が与えられたとする．頂点を $s_i \in \boldsymbol{S}$，辺を (s_i, s_j) $s.t.$ $s_i, s_j \in \boldsymbol{S} \wedge i \neq j \wedge dist(s_i, s_j) \leqq h$ とするような構造を $\boldsymbol{S}$ の連結グラフ $\boldsymbol{G}_s$ と呼ぶ．また，含まれるセンサの数を $\boldsymbol{G}_s$ の $size$ と呼ぶ．

定義 9 ($FOLLOWER$) 連結グラフ $\boldsymbol{G}_s$ が与えられたとき，$size$-k の連結グラフ $\boldsymbol{X} \subseteqq \boldsymbol{G}_s$ と $size$-$(k{+}1)$ の連結グラフ $\boldsymbol{Y} \subseteqq \boldsymbol{G}_s$ について $\boldsymbol{Y}$ が $\boldsymbol{X}$ 上のすべてセンサを含むとき $\boldsymbol{Y}$ を $\boldsymbol{X}$ の $FOLLOWER$ と呼ぶ．

定義 10 ($PARENT$) $size$-k の連結グラフ $\boldsymbol{Y} \subseteqq \boldsymbol{G}_s$ が与えられたとする．$\boldsymbol{Y}$ 内のノードに優先順序を付与し，その優先順位を ν とする．$s \in \boldsymbol{Y}$ とするとき，$\boldsymbol{X} = \boldsymbol{Y} - \{s\}$ が連結グラフとなるような s に対して，ν の中で最も優先順位が高いものを s_p とする．このとき $\boldsymbol{X} = \boldsymbol{Y} - \{s_p\}$ を $\boldsymbol{Y}$ の $PARENT$ と呼ぶ．

センサ集合 $\boldsymbol{S}$ の SCP 探索木は，$\boldsymbol{S}$ の連結グラフ $\boldsymbol{G}$ 内に存在する全ての部分グラフ $\boldsymbol{G}_i$ を重複なく用いて構築される．また，各子ノードは親ノードの $FOLLOWER$ であり，各子ノードは唯一の $PARENT$ をもつ．SCP 探索木は，ルートから深さ優先探索を実施することで一度の探索で連結グラフ内 $\boldsymbol{G}$ のすべての部分グラフ $\boldsymbol{G}_i$ を重複なく探索することができるという特徴を持つ．ベースライン手法では，この特徴を利用しセンサクラスタ内に存在する全ての近傍集合上の属性間の相関を探索する．まず，センサクラスタの連結グラフに対応する SCP 探索木を構築する．SCP 探索木のルートから深さ優先探索を行うことでクラスタ内の全ての近傍集合を一度の探索で重複なく走査することができる．そして，各近傍集合上にて属性間の相関の有無を計算し，属性間の相関が存在する場合に結果を出力することで，センサクラスタ内に存在する全ての属性間の相関を発見する．各近傍集合内の相関の有無は属性間の空間的な共起のサポートで判定する．正の相関を探索する場合，近傍集合内においてセンサの属性に関わらず正の $evoloving$ タイムスタンプ集合同士および負の $evolving$ タイムスタンプ集合同士の積集合を計算する．属性間の空間的な共起のサポートがしきい値 ψ 以上であるならば，近傍集合 $\boldsymbol{Y}$ 上に正の相関があると判定する．一方，属性間の負の相関を探索する場合，近傍集合内において，同じ属性のセンサ間では，正の $evolving$ タイムスタンプ同士および負の $evolving$ タイムスタンプ同士の積集合を計算する．異なる属性のセンサ間では正の $evolving$ タイムスタンプと負の $evolving$ タイムスタンプの積集合を計算する．属性間の空間的な共起のサポートがしきい値 ψ 以上であるならば，近傍集合 $\boldsymbol{Y}$ 上に負の相関があると判定する．なお，しきい値 ψ はユーザ定義で与える．また，定理 1 より，近傍集合 $\boldsymbol{G'}$ 上で属性間の相関が存在しない場合，そのいかなる上位集合 $\boldsymbol{G}$ 上にも属性間の相関は存在しないことが言える．これより，近傍集合内に正および負の相関がどちらも存在しない，あるいは近傍集合の上位集合が存在しない場合に近傍集合の拡張を打ち切り，計算回数の枝刈りを行う．

定理 1 二つの近傍同属性集合を $\boldsymbol{G}_{a_1}, \boldsymbol{G}_{a_2} (a_1 \neq a_2)$ とし，$\boldsymbol{G} = \boldsymbol{G}_{a_1} \cap \boldsymbol{G}_{a_2}$ 近傍集合とする．$\boldsymbol{G}$ の部分集合の内，近傍集合である部分集合を $\boldsymbol{G'} = \boldsymbol{G'}_{a_1} \cap \boldsymbol{G'}_{a_2} \subset \boldsymbol{G}$ とするとき，いかなる $\boldsymbol{G'}$ の属性間の相関 $P_{G'}, N_{G'}$ について，$|P_{a_1,a_2}| \leqq |P'_{a_1,a_2}|$ かつ $|N_{a_1,a_2}| \leqq |N'_{a_1,a_2}|$ である．

証明 $\boldsymbol{G'}$ は $\boldsymbol{G}$ の部分集合であるため，P_{a_1,a_2} に含まれるすべてのタイムスタンプは P'_{a_1,a_2} にも含まれる．ゆえに $|P_{a_1,a_2}| \leqq |P'_{a_1,a_2}|$ である．同様にして $|N_{a_1,a_2}| \leqq |N'_{a_1,a_2}|$ である． □

Algorithm 1 はベースライン手法のアルゴリズムである．まず，line 3 で時系列データの区分線形化を行い，line 4– 13 で時系列データ中から $evolving$ タイムスタンプを抽出する．次に，line 14 でセンサ集合をクラスタリングし，line 15– 17 で各クラスタ内において属性間の相関を探索し結果を出力する．

Algorithm 1 相関探索手法

Input: Two-attribute sensor set $\boldsymbol{S}$, evolving threshold Θ_{a1},Θ_{a2}, distance threshold h, minimum support ψ
Output: All the correlated sensor sets on $\boldsymbol{S}$

```
 1: foreach attribute a that S contains do
 2:     foreach s_i ∈ G_a do
 3:         Segmenting Time Series(s_i)
 4:         for j = 0 → length of T do
 5:             if r_i[t_j] ≧ θ+ then
 6:                 t_j is positive evolving timestamp
 7:             end if
 8:             if r_i[t_j] ≦ θ− then
 9:                 t_j is negative evolving timestamp
10:             end if
11:         end for
12:     end for
13: end for
14: C ← sensor clusters with DBSCAN(S,h,MinPts = 2)
15: foreach c ∈ C do
16:     Correlation Search(c,φ,ψ)
17: end for
```

SCP 探索木は与えられたセンサクラスタに対して概念的に構築されるものである．実際には Algorithm. 2 を初期値 $\boldsymbol{X} = \phi$ として再帰的に実行することで，センサクラスタ内の全ての属性間の相関を発見する．まず，line 1– 3 では連結グラフ X が二つの属性を含む場合，X 上の属性間の相関を出力する．次に，line 4 で連結グラフ $\boldsymbol{X}$ の $FOLLOWER$ を全て選択する．そして，line 5– 12 で近傍集合 $\boldsymbol{X}$ の $PARENT$ に対して属性間の相関の有無を計算し，属性間の相関が存在する場合に Algorithm. 2 を再帰的に繰り返す．

5. 提案手法

本章では提案手法について説明する．まず 5.1 節で提案手法の設計方針について述べ，5.2 節にて提案手法を概説する．提案手法ではパラメータの削減および探索の効率化を行う．それぞれの詳細について 5.3 節および 5.4 節で説明する．

Algorithm 2 Correlation Search

Input: Connected Sensor graph $\boldsymbol{G}$, connected component $\boldsymbol{X} \subseteqq \boldsymbol{G}$, minimum support ψ
Output: The correlated sensor set

1: **if** $\boldsymbol{X}$ contains two kinds of attribute **then**
2: output $\boldsymbol{X}$ and the attribute on $\boldsymbol{X}$
3: **end if**
4: $F(\boldsymbol{X}) \leftarrow FOLLOWER$s of $\boldsymbol{X}$ in the SCP search tree
5: **foreach** $\boldsymbol{Y} \in F(\boldsymbol{X})$ **do**
6: **if** $PARENT$ of $\boldsymbol{Y} = \boldsymbol{X}$ **then**
7: $C_y \leftarrow$ the correlation on $\boldsymbol{Y}$ with Correlation Check($\boldsymbol{Y}$,ψ)
8: **if** C_y is not empty **then**
9: Correlation Search($\boldsymbol{G}$,$\boldsymbol{Y}$,ψ)
10: **end if**
11: **end if**
12: **end for**

5.1 設計方針

相関探索問題ではパラメータ設定の困難性と相関探索の非効率性という二つの課題点が存在する．まずパラメータ設定の困難性について，属性間の相関探索問題では入力パラメータとして以下の三つのしきい値の設定が必要である．

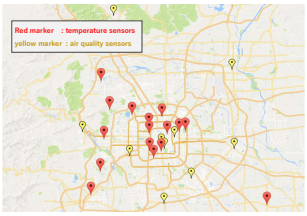
- *evolving* 値 θ: *evolving* タイムスタンプの抽出に用いる測定値の変化量のしきい値．
- 近傍半径 h: 近傍集合の決定に用いる距離のしきい値
- 最小サポート ψ: 近傍集合内に相関があるかどうか判定する際に用いる相関のサポートのしきい値．

この中で，*evolving* 値および近傍半径はセンサ属性またはデータセットごとに適切な値を設定する必要がある．表 1 はセンサ属性ごとの測定値を示す．測定値のスケールはそれぞれ異なるため，異なる属性の時系列データに対して同一の *evolving* 値を用いて *evolving* タイムスタンプの判定を行うことは適切ではない．各属性ごとにそれぞれ個別に *evolving* 値を設定する必要があるが，適切な値を設定することは容易ではない．また図 2 はセンサの設置位置を示す．地域やデータセットによってセンサの地理的な設置間隔はそれぞれ異なるため，データセットごとに近傍半径を適切に調整する必要があるが，適切な値を設定する作業はユーザにとって大きな負担となる．

表 1　センサ属性ごとの測定値の例

センサ属性	1 時間ごとの測定値						
	···	10:00	11:00	12:00	13:00	14:00	···
気温 [°C]	···	18.5	19.0	19.2	20.3	21.4	···
照度 [lux]	···	898.1	5044.6	6502.3	5774.9	16726	···
騒音 [dB]	···	50.9	55.0	59.7	52.3	58.0	···

次に探索の非効率性について，属性間の相関探索問題では多種類の属性を含むセンサ集合内から，二つの属性を含む近傍集合に対して相関の有無を計算する．ベースライン手法では相関探索にセンサ属性を考慮しておらず，二つの属性を含む近傍集合の他に単一種類のみの属性を含む近傍集合についても相関の有無を計算する．図 3 はあるセンサ集合とその連結グラフを示す．ベースライン手法では近傍集合 $X = \{s_3, s_4, s_5\}$ について

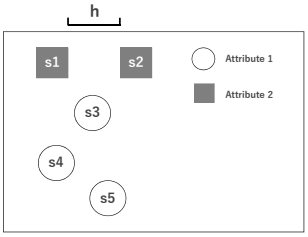


(a) 中国，北京

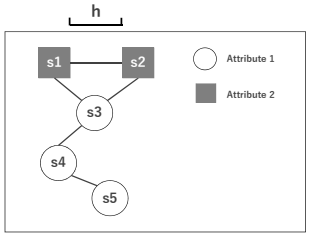
(a) スペイン，サンタンデル

図 2　各地域におけるセンサの設置位置

相関の有無を計算するが，相関探索問題では単一種類の属性を含む近傍集合の相関を確かめる必要はない．ベースライン手法ではこのような必要のない近傍集合についても相関の有無を計算するため相関探索問題において非効率的な手法であるといえる．また，三つ以上の属性を含む近傍集合も対象としていないため入力するセンサ集合は二つの属性を含むセンサ集合に限定される．



(a) センサ集合 $\boldsymbol{S}$ と近傍半径 h



(a) センサ集合 $\boldsymbol{S}$ の連結グラフ $\boldsymbol{G}$

図 3　センサ集合 $\boldsymbol{S}$ と連結グラフ $\boldsymbol{G}$

本稿で提案するフレームワークでは，*evolving* 値および近傍半径の二つの入力パラメータを手法中で自動的に決定することでパラメータ設定の困難性を解決する．また，多種類の属性を含むセンサ集合内から二つの属性を含む近傍集合のみを探索する探索アルゴリズムを提案することにより，相関探索問題に不要な計算を省略し計算効率を向上させる．加えて，新たに提案する探索アルゴリズムでは，多種類の属性を含むセンサ集合内に存在する近傍集合の内，二つの属性を含む近傍集合のみが相関の有無を計算する対象となるため，入力として多種類の属性を含むセンサ集合を考えることができる．

5.2 MISCELA

本稿で新たに提案する属性間の相関探索フレームワーク MISCELA では，五つのステップで属性間の相関を探索する．ベースライン手法と比較して異なる点は以下の三点である．まず，MISCELA では手法中で各属性ごとに *evolving* 値を自動決定する．さらに，クラスタリングアルゴリズとして OPTICS [7] を利用することで，近傍半径を手法中で決定する．また，相関探索には本稿で新たに定義する SCA(Spatial Correlation Attribute) 探索木を利用する．

Algorithm. 3 は MISCELA のアルゴリズムを示す．MISCELA では入力として多種類のセンサを含むセンサ集合と二つの入力パラメータをとる．line 2 では *evolving* 値を自動決定する．ここで k は MISCELA で新たに追加した入力パラメータであり，時系列データ中における *evolving* タイムスタンプ数

の割合を示す．line 15 では OPTICS を用いてクラスタリングを行う．line 17 では本稿で新たに定義する SCA 探索木を利用して，各クラスタ内における属性間の相関を効率的に探索する．

Algorithm 3 MISCELA

Input: Multi-attribute sensor set $\boldsymbol{S}$, evolving ratio k, minimum support ψ
Output: All the correlated sensor sets on $\boldsymbol{S}$

```
1: foreach attribute a that S contains do
2:     Θ_a ←Evolving Threshold Estimation(G_a,k)
3:     foreach s_i ∈ G_a do
4:         Segmenting Time Series(s_i)
5:         for j = 0 → length of T do
6:             if r_i[t_j] ≧ θ_a^+ then
7:                 t_j is positive evolving timestamp
8:             end if
9:             if r_i[t_j] ≦ θ_a^- then
10:                t_j is negative evolving timestamp
11:            end if
12:        end for
13:    end for
14: end for
15: C ← sensor clusters with OPTICS(S,MinPts = 2)
16: foreach c ∈ C do
17:    2-Attribute Correlation Search(c,X,ψ)
18: end for
```

5.3 パラメータの自動決定

MISCELA では，パラメータ調整の困難性を解決するために *evolving* 値および近傍半径の二つの入力パラメータを手法中で自動決定する．*evolving* 値の決定は新たに追加する入力パラメータ k を用いて行う．*evolving* タイムスタンプの数が最大となるのは $\Theta = (0, 0)$ の場合であり，ここから各属性ごとに *evolving* タイムスタンプ数が最大値の k% になるように *evolving* 値を決定する．なお，MISCELA では $|\theta^+| = |\theta^-|$ となるように *evolving* 値を決定する．

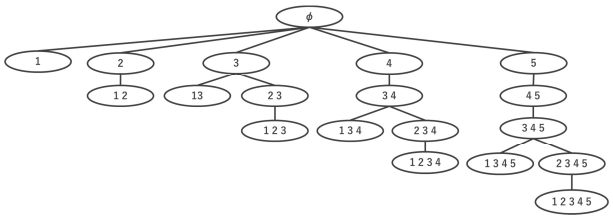
近傍半径の決定にはクラスタリングアルゴリズム OPTICS を利用する．OPTICS は DBSCAN を拡張した手法であり，各オブジェクトにおいて *Reachability Distance* と呼ばれる最近隣オブジェクトまでの距離を計算し，リストとして保持しておく．MISCELA では全センサに対して *Reachability Distance* を計算した後，その最大値の $\frac{1}{2}$ を近傍半径とすることで，近傍半径の自動決定を行う．

5.4 SCA 探索木

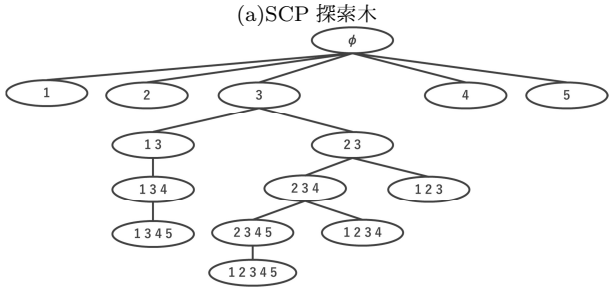
本稿では，多種類の属性を含むセンサ集合から効率的に属性間の相関を探索するための木構造 SCA 探索木を提案する．SCP 探索木ではセンサ属性を考慮しないため，ノードに単一属性のみを含む近傍集合が存在する．ルートからの深さ優先探索によって，単一属性のみを含む近傍集合に対しても相関の有無を計算するが，相関探索問題では不要な計算である．そこで，SCA 探索木では与えられた連結グラフから，連結グラフ内の各センサと二つの属性を含む部分グラフのみを用いて木構造を構築する．これにより，与えられたセンサ集合からの二つの属性を含む近傍集合のみに対して相関の有無を計算することができる．まず，2-Attribute $PARENT$ という連結グラフ間の関係を新たに定義する．

定義 11 (2-Attribute $PARENT$) *size-k* の連結グラフ $\boldsymbol{Y} \subseteqq \boldsymbol{G}_s$ が与えられたとする．$\boldsymbol{Y}$ 内のノードに優先順序を付与し，その優先順位を ν とする．$s \in \boldsymbol{Y}$ とするとき，$\boldsymbol{X} = \boldsymbol{Y} - \{s\}$ が二つのセンサ属性を含む連結グラフとなるような s に対して，ν の中で最も優先順位が高いものを s_p とする．このとき $\boldsymbol{X} = \boldsymbol{Y} - \{s_p\}$ を $\boldsymbol{Y}$ の *2-Attribute PARENT* と呼ぶ．

SCA 探索木は SCP 探索木と異なり，連結グラフ内の各ノードと二つの属性を含む連結グラフのみを用いて構成される．各子ノードは親ノードの $FOLLOWER$ であり，各子ノードは唯一の 2-Attribute $PARENT$ をもつ．SCA 探索木は，ルートから深さ優先探索を実施することで, 一度の走査で連結グラフ内の各ノードと二つの属性を含む部分グラフのみを重複なく探索することができるという特徴を持つ．MISCELA では，この特徴を利用しセンサクラスタ内に存在する全ての近傍集合内の属性間の相関を効率的に発見する．まず，センサクラスタの連結グラフに対応する SCA 探索木を構築する．SCA 探索木のルートから深さ優先探索を行うことでクラスタ内の二つの属性を含む近傍集合を一度の探索で重複なく走査することができる．そして，各近傍集合上にて属性間の相関の有無を計算し，属性間の相関が存在する場合に結果を出力することで，センサクラスタ内に存在する全ての属性間の相関を発見する．図 4 は図 3 の連結グラフの SCP 探索木および SCA 探索木を示す．ここで，$\nu = \{1 \to 2 \to 3 \to 4 \to 5\}$ とする．



(a)SCP 探索木



(b)SCA 探索木

図 4　SCP 探索木と SCA 探索木

SCA 探索木は SCP 探索木と同様に，与えられたセンサクラスタに対して概念的に構築されるものである．実際には Algorithm. 4 を初期値 $\boldsymbol{X} = \phi$ として再帰的に繰り返すことで，クラスタ内の全ての属性間の相関を発見する．まず，line 1– 3 では連結グラフ X が二つの属性を含む場合，X 上の属性間の相関を出力する．また，ine 5– 12 では近傍集合 $\boldsymbol{X}$ の 2-Attribute $PARENT$ に対して，属性間の相関の有無を計算し，属性間の相関が存在する場合に Algorithm. 4 を再帰的に繰り返す．

Algorithm 4 2-Attribute Correlation Search

Input: Connected graph $\boldsymbol{G}$, connected component $\boldsymbol{X} \subseteqq \boldsymbol{G}$, minimum support ψ

Output: The correlated sensor set

```
if |X| ≧ 2 then
    output X and the attribute on X
end if
F(X) ← FOLLOWERs of X in the SCA search tree
foreach Y ∈ F(S) do
    if 2-Attribute PARENT of Y = X then
        C_y ← the correlation on Y with Correlation Check(Y,ψ)
        if C_y is not empty then
            2-Attribute Correlation Search(G,Y,ψ)
        end if
    end if
end for
```

6. 評価実験

6.1 実験環境

本稿ではスペイン，サンタンデル市に設置されている環境センサ用いて実験を行う．センサ属性は気温，照度，騒音，交通量，湿度の五種類を用いるとし，詳細は表 2 に示す通りである．また，データの測定期間は 2016 年 3 月 1 日から 9 月 30 日までの 7 か月間である．図 5 にセンサの設置位置を示す．赤色，青色，白色，黄色，緑色のピンはそれぞれ気温センサ，湿度センサ，騒音センサ，照度センサ，交通量センサである．

表 2　データセット

属性	単位	センサ数
気温	[°C]	297
照度	[lux]	181
騒音	[dB]	32
交通量	[%]	31
湿度	[%]	10

図 5　スペイン，サンタンデル市の環境センサ

また，アルゴリズムは全て C++を用いて実装し，実験は Intel Corei7-6700K 2.4GHz の CPU，16GB の RAM を搭載したコンピュータ上で行う．

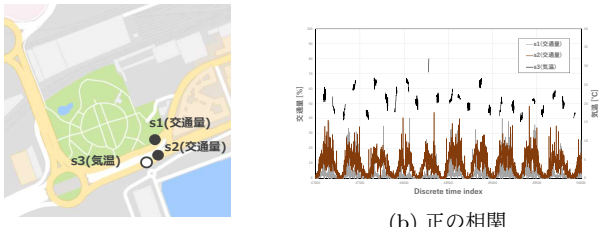
6.2 実験結果

本節では実験結果について述べる．まず，MISCELA を用いて実データ中から探索した属性間の相関について考察する．次に，入力するセンサ集合やパラメータを変更しながら，ベースラインと MISCELA の実行時間を比較し，探索の効率性について考察する．最後に，MISCELA の特徴の一つであるパラメータの自動決定について実行時間の評価を行う．

6.2.1 属性間の相関の例

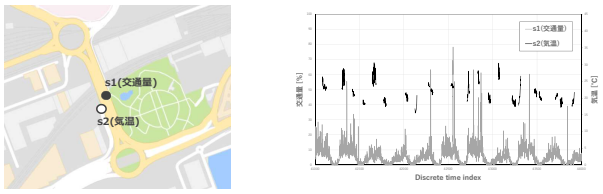
本実験では気温，照度，騒音，交通量，湿度の五属性を含むセンサ集合を入力とし，属性間の相関探索を行う．なお，時系列データのサンプリング間隔を $\Delta T = 5$[分]，*evolving* 率を $k = 50$，最小サポートを $\psi = 50$ とする．

図 6 は入力したセンサ集合から発見された交通量と気温の正の相関を示す．s_1 および s_2 は交通量センサであり，s_3 は気温センサである．図 6(a) からわかるようにこれらのセンサは空間的に近い位置関係にある．また図 6(b) より同属性間で同変化し，属性間でも同変化している部分時系列が存在することがわかる．正の相関を観測した原因として，通勤や通学による交通量の増加と日の出による気温の上昇がどちらも朝に発生する現象であることなどが考えられる．



(a) センサの位置関係　　(b) 正の相関

図 6　交通量と気温の正の相関



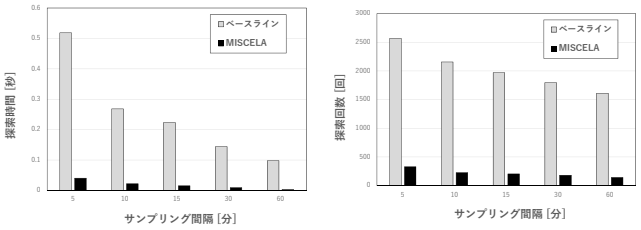
(a) センサの位置関係　　(b) 負の相関

図 7　交通量と気温の負の相関

図 7 は入力したセンサ集合から発見された交通量と気温の負の相関を示す．s_1 は交通量センサであり，s_2 は気温センサである．図 7(a) からわかるようにこれらのセンサは空間的に近い位置関係にある．また図 7(b) より同属性間で同変化し，属性間では逆変化している部分時系列が存在することがわかる．なお，s_1 と s_2 の間には正の相関も観測されている．負の相関を観測した原因として，帰宅ラッシュや夜間における駅前の交通量の増加と日の入による気温の下降が同時に発生したことなどが考えられる．

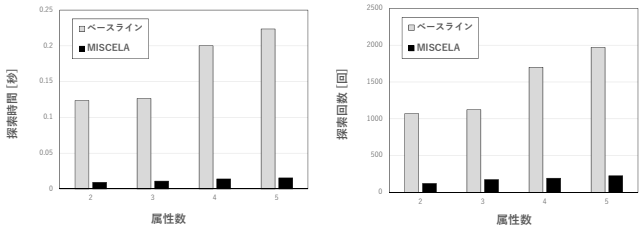
6.2.2 効率性

次に，MISCELA で用いる探索アルゴリズムの効率性について評価を行う．本実験では時系列データのサンプリング間隔 ΔT および入力属性数 M_a をそれぞれ変化させた場合について，ベースラインと MISCELA における探索アルゴリズムの実行時間を比較する．なお入力パラメータは $k = 50$，$\psi = 50$ とする．実験結果をそれぞれ図 8 および図 9 に示す．



(a) 探索時間　　(b) 探索回数

図 8　サンプリング間隔の影響



(a) 探索時間　　(b) 探索回数

図 9　入力属性数の影響

サンプリング間隔を短くすることで時系列データ長が増長し，各近傍集合上における相関の有無の計算時間が増加したため，探索時間および探索回数が増加したと考えられる．また，入力属性数が増加することで探索の組合せ数が増加し，探索時間および探索回数が増加したと考えられる．MISCELA はサンプリング間隔 ΔT を変化させた場合において探索時間を最大で 97% 短縮する．

6.2.3　パラメータ決定時間

最後に，MISCELA の特徴の一つであるパラメータの自動決定について実行時間の評価を行う．本実験では時系列データのサンプリング間隔 ΔT を変化させた場合の *evolving* 値の決定時間および，入力センサ数 M_s を変化させた場合のクラスタリング時間を測定する．入力パラメータは $k = 50$，$\psi = 50$ とする．実験結果をそれぞれ図 10 および図 11 に示す．

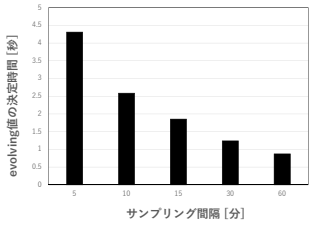


図 10　evolving 値の決定時間におけるサンプリング間隔の影響

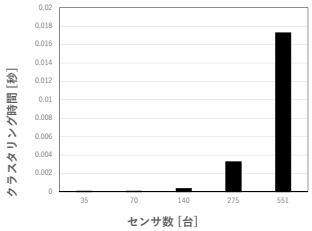


図 11　クラスタリング時間における入力センサ数の影響

MISCELA では *evolving* 値を決定する際に時系列データの全点を複数回走査するため，*evolving* 値の決定に要する時間はデータの時系列長に比例する．実験結果より，実際に時系列の長さに比例して *evolving* 値の決定に要する時間が増加していることがわかる．また，クラスタリングアルゴリズム OPTICS の時間計算量は入力センサ数を M_s とするとき，$O(M_s^2)$ であることが知られている．一方，近傍半径の決定には全センサの *Reachability Distance* を用いるため，計算量は $O(M_s)$ である．すなわち，クラスタリングに要する時間は入力センサ数の二乗に比例すると考えられる．実験結果より，実際にクラスタリング時間は入力センサ数に従って指数的に増加していることがわかる．

7.　結　　論

本稿では，多種類の属性を含むセンサ集合から二属性間の相関を発見する属性間の相関探索問題を新たに定義し，相関探索フレームワーク MISCELA を提案した．MISCELA では，相関探索に必要な入力パラメータの中で設定が困難である *evolving* 値と近傍半径を手法中で自動的に決定する．これにより，センサ属性やデータセットごとにパラメータを手動で調整する作業が不要となる．また，多種類の属性を含むセンサ集合の中から二つの属性を含む近傍集合のみを重複なく探索することにより効率的な相関探索が可能となる．本稿では実センサデータを用いて実験を行った．その結果，データセットの中からいくつかの相関をもつ属性を発見した．また，MISCELA はベースライン手法と比較して属性間の相関探索問題において効率的な探索を実施できるという結果が得られた．

今後の課題としては時間的なずれを考慮した相関探索手法の検討が考えられる．本稿では同時変化のみを対象として相関探索を行ったが，実世界においては，気温が上昇した少し後に交通量が増加するなどといった時間的なずれをもった相関が存在する可能性も高い．相関探索に時間的なずれを考慮すると計算パターンが増加し計算量が膨大となる．そのため時系列上の特定パターンを効率的に探索する探索アルゴリズムが必要であると考えられる．

謝　　辞

本研究は科学研究費 (16H01722) の支援によって行われた．ここに記して謝意を表す．

文　　献

[1] Shashi Shekhar, Zhe Jiang, Reem Y Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information*, Vol. 4, No. 4, pp. 2306–2338, 2015.

[2] Chao Zhang, Yu Zheng, Xiuli Ma, and Jiawei Han. Assembler: efficient discovery of spatial co-evolving patterns in massive geo-sensory data. In *Proceedings of the ACM SIGMOD*, pp. 1415–1424, 2015.

[3] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Processings of thte IEEE ICM*, pp. 289–296, 2001.

[4] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2011.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceefings of the ACM SIGKDD*, pp. 226–231, 1996.

[6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 24, No. 5, pp. 603–619, 2002.

[7] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *Processings of the ACM SIGMOD*, pp. 49–60, 1999.

[8] Yun Cheng, Xiucheng Li, and Yan Li. Finding dynamic co-evolving zones in spatial-temporal time series data. In *Processings of the ECML PKDD*, pp. 129–144, 2016.