RF-007

同期とグラフを用いたクラスタリング手法の提案と評価

A clustering method using graph and synchronization

速水 雄太郎† 菅原 俊治†
Hayamizu Yutaro Sugawara Toshiharu

1 はじめに

近年、大量のデータが計算機で処理され、今後そのデータ量は情報化社会の発展に伴い増加する傾向にある。そのため、大量のデータの中から何らかの意味をもつ情報を抽出することが現在の社会において必要となる。そのような現状において、クラスタリングは教師なしデータ分類の重要な一手法として用いられており、情報科学のみならず経済学や自然科学、社会科学、言語学など様々な分野で利用されており、各分野において基礎的な役割を持っている。

しかし、利用されている従来のクラスタリング手法では任意形状のクラスタの検出が難しく、また、データに含まれるノイズに対して脆弱であるなどの欠点も抱えている。実世界のデータには、明確にクラスタリングできるものばかりでなく、ノイズを含んだデータが数多く存在する。また、データ自体がいくつの種類に分類できるか分からない場合も多くある。そのため、現実のデータに使用するためのクラスタリングアルゴリズムには、ノイズに対する安定性や、クラスタの数を自動で推定する機能などが求められてくる。

一方、複雑系の分野では、生物などに見られる自然界の同期現象を利用することによって、外部からの影響に対して安定性の高いシステムを構築できることが分かってきている。そのため、同期現象を利用し、クラスタの大きさや数に依存しないクラスタリングアルゴリズムを作ろうという動きが生まれている。このような観点から提案されたクラスタリングアルゴリズムとして、複雑系における結合振動子を用いたものがある [1]。しかし、この手法を含む多くのクラスタリングアルゴリズムは、クラスタの形状が球状または楕円状であることを仮定しているため、実世界にあるクラスタ形状が不定形なデータに対してのクラスタリングが困難であった。本論文ではこの手法を拡張し、任意形状のクラスタ分割が可能なアルゴリズムを提案する。また、ベンチマーク用の人工データセットと、実データとして UCI Machine Learning Repository[7] のデータを用いて、その性能を評価することを目的とする。

2 関連研究

2.1 同期現象

自然界には集団を構成する個体が影響しあい、自己組織的に同期する現象が多く見られる。例えば蛍は、互いに離れている時は各々の周期によって独立に発光している。しかし、群れになると発光により近隣の蛍同士で互いに刺激を与え合い、周期を調整して、群れ全体が同期して発光するようになる。Mirollo と Strogatz はこれらの現象をモデル化し、パルス結合振動子モデル（pulse-coupled oscillators model）として紹介した [2]。このモデルでは各振動子は全て同じ形をしており、振動子間は発火により影響を受ける。言い換えると、発火した振動子は他の振動子に興奮や発火、または抑制する効果を与える。パルス結合振動子モデルは、近隣の振動子との局所的な相互作用のみで全体の同期を達成するため、集中制御の必要がなく、センサーネットワークの時刻同期手法などにも利用されてる [3]。

2.2 クラスタリング

クラスタリングのアルゴリズムは大きく分けて二種類ある。一つはウォード法などに代表される階層的クラスタリングである。階層的クラスタリングは最初、各データを一つのクラスタと見なし、類似度の高いクラスタを次々に統合して最終的な結果を得る。一般的に階層的クラスタリングは、良い分類結果を得られるが計算に時間がかかるため、大規模なデータのクラスタリングに利用することは難しい。二つ目は非階層的クラスタリングと呼ばれ、K-means 法が有名である。非階層的クラスタリングではクラスタの良さを示す目的関数を定義し、その関数を最適化するようにクラスタを探索する。多くの場合、探索が局所最適解に陥るため、クラスタリング結果が初期値に依存するという欠点を持つ。しかしながら、解の探索は比較的簡単に計算でき、階層的クラスタリングに比べ高速に結果を得られることなどから大規模データのクラスタリングに用いられることが多い。よって本節では、より実用性の高い非階層的クラスタリングに主眼を置き、関連するクラスタリング手法を説明する。

任意形状のクラスタ分割が可能な手法の一つにグラフを用いたものがある。Urquhart は幾何グラフの一種である Gabriel グラフや相対近傍グラフを用いた手法を提案した [4]。この方法では各データをグラフで結んだ後、長い辺を取り除くことでクラスタ分割を行う。

また、パルス結合振動子を用いたクラスタリング手法が Rhouma と Frigui によって提案された [1]。パルス結合振動子には、似た振る舞いをする振動子同士がそれぞれ異なった小グループを作って同期するという現象があり、哺乳類の脳の視覚皮質などにおいて同様の現象がみられる。[1] のモデルにおいても、各振動子は同じグループ内の他の振動子と同期して発火し、グループそのものは他のグループと一定の位相差をもって発火する。グループの構造がどうなるかは振動子間の結合強度関数の選択に依存し、このモデルでは振動子間の結合強度をその相対距離に基づいて定義する。また、同期を用いたクラスタリングは、振動子同士の自己組織化により任意のクラスタ数を見つけることが出来る。自己組織化によりモデルが安定化するだけでなく、クラスタリングと自己組織化の相乗作用によって計算複雑度が顕著に減少する。しかしながら、このモデルでは同じ平均サイズのグループを生み出す傾向があり、必ずしもクラスタリングアルゴリズムとして良い結果を得られるとは限らない。

そこで、本論文では先に紹介したグラフを用いた手法とパルス結合振動子を用いた手法を組み合わせることで、二つのクラスタリングの利点である、任意形状の分割可能であり、安定性に優れ、計算複雑度も低いと云った性質を併せ持つクラスタリングアルゴリズムを提案する。

†早稲田大学理工学術院基幹理工学研究科情報理工学専攻
y.hayamizu@isl.cs.waseda.ac.jp

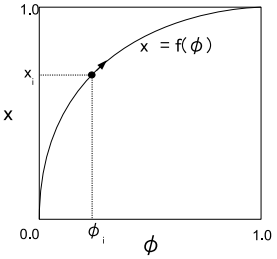

図 1: パルス結合振動子のモデル

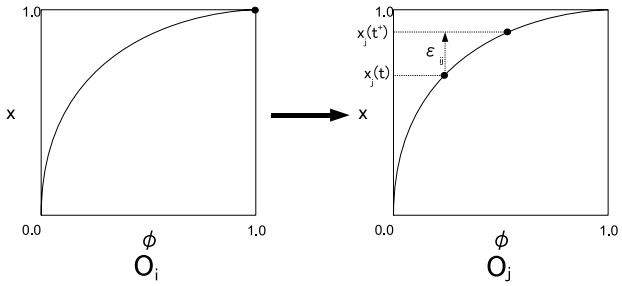

図 2: 発火の影響

3 パルス結合振動子

N 個の振動子 $\mathbf{O} = \{O_1, \cdots, O_N\}$ を考える。振動子 O_i は位相 $\phi_i \in [0,1]$ と状態値 $x_i \in [0,1]$ を持ち、これらは以下の式で関係付けられる。

$$x_i = f_i(\phi_i) \tag{1}$$

$f_i : [0.1] \rightarrow [0,1]$ は任意の単調増加関数で、$f_i(0) = 0$、$f_i(1) = 1$ である。式 (1) を図示したものが図 1 である。位相 ϕ_i は時間と共に増加し、状態値 x_i も式 (1) に従い増加する。$x_i = 1$ となった時に振動子 O_i は発火し、状態値 x_i、位相 ϕ_i は共に 0 にリセットされる。これを 1 セットとし振動子は、増加→発火→リセットを繰り返す。本研究では [1] に倣い、式 (1) の単調増加関数 f_i を次式とする。

$$\forall i, \quad f_i(\phi_i) = \frac{1}{b}\ln[1 + (e^b - 1)\phi_i] \tag{2}$$

$b > 0$ は関数の増加具合を表すパラメータで、b が大きくなるにつれて関数値の上昇具合は急激になる。

また、発火した振動子 O_i は他の振動子 O_j を刺激し、状態値 x_j は結合強度 ε_{ij} 分増加する。

$$x_j(t^+) = B(x_j(t) + \varepsilon_{ij}) \tag{3}$$

この時、合わせて位相 ϕ_j も式 (1) に従い増加する。ここで、関数 B は状態値 x_j を [0,1] の範囲に限定する為に使われ、式 (4) で表す。

$$B(x) = \begin{cases} x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x < 0 \\ 1 & \text{if } x > 1 \end{cases} \tag{4}$$

図 2 は式 (3) の意味を図示したものである。振動子 O_i と振動子 O_j は結合強度 ε_{ij} の値によって刺激、または抑制し合う。振動子 O_j が振動子 O_i からの発火の刺激を受けた結果、状態値 x_j が 1 に達した場合、振動子 O_j もまた発火する。このとき振動子 O_i と振動子 O_j は同期したと見なされる。式 (3) での結合強度 ε_{ij} は次式で記述する。

$$\forall i \neq j, \quad \varepsilon_{ij} = \begin{cases} C_E(1 - (\frac{d_{ij}}{\delta_0})^2) & \text{if} \quad d_{ij} \leq \delta_0 \\ 0 & \text{if} \quad d_{ij} > \delta_0 \end{cases} \tag{5}$$

d_{ij} は振動子 O_i と振動子 O_j の距離である。δ_0 は振動子 O_i と振動子 O_j の類似度を定義するパラメータで、$d_{ij} < \delta_0$ ならば振動子 O_i と振動子 O_j は類似していると見なし、$d_{ij} > \delta_0$ ならば類似していないと見なす。C_E は振動子同士が類似していた場合に与える正の結合強度の最大値を決めるパラメータである。正の結合強度 ε_{ij} の強さは d_{ij} と δ_0 の大きさに依存する。もし、d_{ij} と δ_0 の値が非常に近ければ類似性の有無が疑わしいため、弱い結合強度となる。逆に、$d_{ij} \ll \delta_0$ のように両者の差が大きければ類似性の有無は明らかなため強い結合強度となる。通常、各パラメータは $b = 3.0, C_E = 1.0$ の定数とする。各パラメータの効果については後述の 4.3 節で議論する。

本論文において、振動子は全て同じ周期と状態の関数を持っていると仮定しているが、異なった周期や位相と状態の関数を持った振動子間においても任意の初期状態から同期することが証明されている [2]。また、全ての振動子が互いに刺激を与え合う関係になくとも、間接的な刺激の伝播により、振動子集合全体は同期に達する。

4 アルゴリズムの提案

4.1 提案アルゴリズムの概要

本クラスタリングアルゴリズムの特徴は、データをグラフを用いて大域的に関係付けた後、局所的な自己組織化をパルス結合振動子を用いて行うという二つのプロセスを組み合わせた点にある。まず、データの大域的な関係を求めるために、データ間の距離からデータの相対近傍グラフを作成する。相対近傍グラフとは幾何グラフの一種で、次のように定義する。まず、点集合 $S = \{O_1, O_2, \cdots, O_n\}$ があるとき、2 つの点 O_i, O_j の距離を $d(O_i, O_j)$ とする。このとき、次式を満たした O_i, O_j を辺で結ぶことで得られるグラフを相対近傍グラフという。

$$d(O_i, O_j) \leq \min_{k(\neq i,j)} \max\{d(O_i, O_k), d(O_j, O_k)\} \tag{6}$$

このようにデータ間をグラフで結ぶことで、同じクラスタに属するデータ同士は密なグラフ構造を構成する。一方で異なるクラスタに属するデータ間では疎なグラフ構造しか作れないため、辺の粗密により大域的にデータを分類できる。また、相関の低いデータ同士は辺を結ばないためデータ間の関係を簡略化できる。次に、局所的な自己組織化のプロセスでは、相対近傍グラフによって限定されたデータ間の関係に着目する。各データにパルス結合振動子モデルを適応し、データ間の距離に応じて刺激を与える。刺激を与えられたデータ同士は同期によって徐々に組織化し、刺激を与えられないデータは孤立する。最終的に同期したデータを抽出することで、このクラスタリングアルゴリズムの結果を得る。

4.2 アルゴリズム

4.2.1 アルゴリズムの流れ

提案アルゴリズムを RCON(Ripple-Carry Oscillators Network) と呼ぶ。RCON を説明するために図 3 にアルゴリ

```
Procedure Clustering(S)
Input: S:データ集合;
Output: C:クラスタ集合;
    データ間距離を配列 D に格納;
    for(全てのデータの組み合わせに対して)
        式 (6) を用いて相対近傍グラフの作成;
        if(グラフが作成された)
            式 (5) を用いて結合強度の計算;
        end if
    end for
    振動子を初期化 ;
    while(同期が安定する)
        次に発火する振動子 O_i を判別;
        振動子を発火点まで推移;
        同期処理 Synchronize(O_i);
        同期した振動子を判別してリセット;
    end while
return C;
end Procedure
```

図 3: アルゴリズム全体の流れ

ズムの流れを示す。RCON ではクラスタリングを行うデータ集合 S が与えられると、各データ間の距離を求め、行列 D に格納する。今回は簡単のためにユークリッド距離を用いた。次に、行列 D を基に式 (6) を用いてデータ集合 S の相対近傍グラフを作る。同時に、式 (5) を使い、相対近傍グラフの辺で結ばれたデータ同士の結合強度 ε を計算する。これにより無駄な結合強度の計算を省くことが出来る。ここまででグラフによるデータ間の大まかな関係付けは終わり、続けて小さな範囲での自己組織化に移る。データ毎に割り当てられた各振動子の位相をランダムに初期化し、その中から次に発火する振動子 O_i の位相 ϕ_i、つまり全振動子の中で最大の位相を取得する。次に、全ての振動子の位相に $1-\phi_i$ を加え、振動子 O_i が発火する状態にまで推移させる。振動子 O_i が発火したら、相対近傍グラフで結ばれた他の振動子に対し同期処理を行う。もし、相対近傍グラフで結ばれた先の振動子 O_{j} が結合強度 ε_{ij} により発火した場合には同じように同期処理を行う。同期処理については後述の 4.2.2 節でも説明する。同期処理が終了したら同期した振動子をリセットし、再び次に発火する振動子を探す。これら一連の処理を同期が安定するまで繰り返し、最終的に同じ位相で同期した振動子の集合を同一クラスタに属すると見なしてクラスタリングを完了する。

4.2.2 同期処理

発火した振動子 O_i と相対近傍グラフで結ばれた全ての振動子 O_j に対して、同期処理を再帰的に行う。図 4 は同期処理のアルゴリズムを示したものである。まず、処理が無限ループに陥らないように、この一連の同期処理中に O_j が発火したかを確認する。もし発火していれば処理を終了し、次の振動子に対して処理を行う。発火していなければ同期を行う一連の計算に移る。同期の計算では、まず位相 ϕ_{j} から式 (2) を用いて状態値 x_j を求める。次に、状態値 x_j を式 (3) に当てはめ、新しい状態値 x_j^+ を求める。なお、式 (3) で使う結合強度 ε_{ij} は、相対近傍グラフ作成時に式 (5) を使い計算したものを用いる。新しく求めた状態値 x_j^+ は式 (2) の逆関数を使い、位相値 ϕ_j^+ に戻す。ここで再び振動子 O_j が発火したかを判定し、発火していれば新たに O_j に対して同期処理を行い、発火していなければ処理を終了する。このように再帰的に同期処理を行うことによって振動子 O_i から発生する同期の影響を網羅的に得る。

```
Procedure Synchronize(O)
Input: O_i:発火した振動子 i;
    振動子 O_i を発火履歴に追加;
    for(振動子 O_i とグラフで結ばれた
        全ての振動子 O_j に対して)
        if(振動子 O_j が発火していない)
            式 (2) を用いて関数値を再計算;
            式 (3) を用いて重み付け;
            式 (2) の逆関数を用いて位相を再計算;
            if(振動子 O_j が発火した)
                Synchronize(O_j);
            end if
        end if
    end for
end Procedure
```

図 4: 同期処理のアルゴリズム

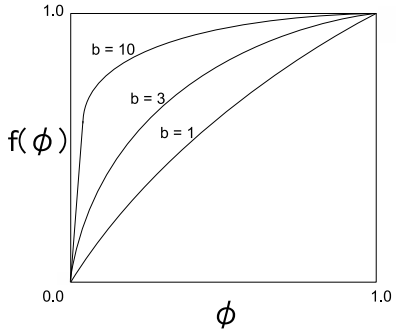

図 5: b と $f(\phi)$ の変化

4.3 各パラメータの効果

4.3.1 b, C_E の効果

振動子の状態値を求める式 (2) のパラメータ b と、結合強度の式 (5) のパラメータ C_E はどちらもクラスタリングの速度を決める。図 5 にパラメータ b と状態値 $f(\phi)$ の関係を示す。パラメータ b が大きいと状態値 $f(\phi)$ の増加の仕方は急激に大きくなり、振動子は少しの結合強度が加算されただけでも発火するようになる。また、パラメータ C_E が大きくなる場合は、結合強度が大きくなり、同様に振動子が発火しやすくなる。通常、クラスタリングの速度を遅くする必要はないが、これらのパラメータを調節すれば、クラスタリングの変化の様子を観察できる。

4.3.2 δ_0 の効果

結合強度を得るための式 (5) で使うパラメータ δ_0 はクラスタリングの精度を決めるパラメータである。δ_0 が小さいほどデータは細かいクラスタに分割され、δ_0 が大きいほどデータは大雑把に分割される。図 6 はパラメータ δ_0 を変えてクラスタリングを行った結果である。分割されたクラスタ数を k で表すと、(a) の $\delta_0 = 0.03$ の場合、データは細かいクラスタに分割され、$k = 78$。(b) の $\delta_0 = 0.08$ の場合にはクラスタ数が減り、$k = 8$ に分割される。更に (c) の $\delta_0 = 0.12$ の場合では、$k = 1$ となる。実際にクラスタリングを行う際には δ_0 を適切に設定する必要がある。

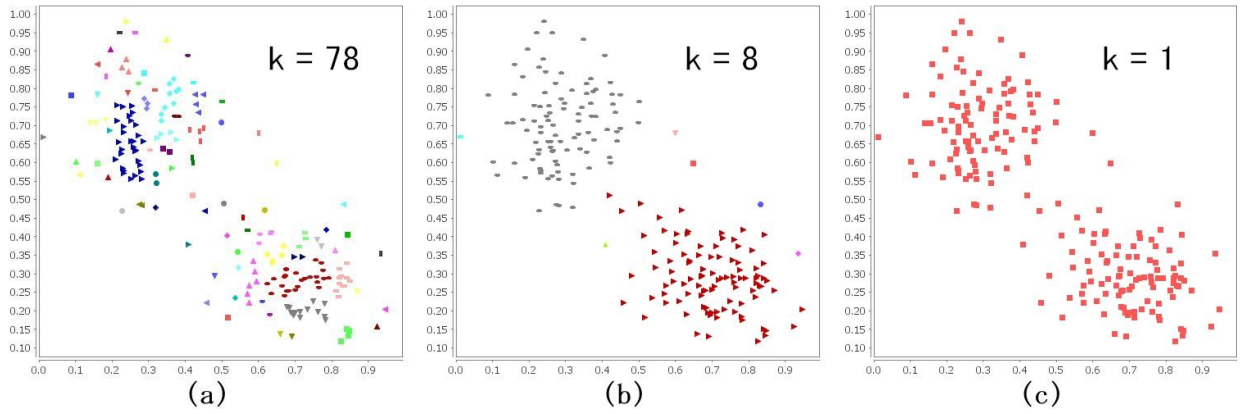

図 6: δ_0 の効果 ((a)$\delta_0 = 0.03$、(b)$\delta_0 = 0.08$、(c)$\delta_0 = 0.12$, k はクラスタ数)

表 1: Jaccard 係数の分類法

		正解セット	
		ペアが同じ	ペアが異なる
クラスタリング後の結果	ペアが同じ	a	b
	ペアが異なる	c	d

表 2: 人工データセットの概要

データセットの種類	データ数	正解クラスタ数
Circle	350	3
Amoeba	199	4

5 性能評価

5.1 評価手法

クラスタリングの評価については様々な手法が提案されている。クラスタリングの評価手法は大きく分けて外的基準を用いたものと、内的基準を用いたものの 2 種類がある。外的基準を用いたものは、クラスタリングを行うデータセットに予め正解が用意されており、クラスタリング結果とその正解を比較して評価を行う。一方、内的基準を用いた評価手法では、クラスタリング結果のみから評価を行う。[5] では、2 種類の外的基準を用いた評価手法と、30 種類の内的基準を用いた評価手法でクラスタリング結果を比較している。本論文では [5] で用いられている外的評価手法の一つ、Jaccard 係数を用いてクラスタリング結果を評価する。

Jaccard 係数は表 1 に表すような簡単な分類で計算することができる。データセット中の任意の 2 つのデータの組み合わせ $n(n-1)$ 個に対し、クラスタリング前の正解セットで同じクラスタに分類された否か、クラスタリング後に同じクラスタに分類されたか否かの 4 通りの組み合わせで分類する。例えば、あるデータのペアがクラスタリング前は同じクラスタに分類され、クラスタリング後に異なるクラスタに分類されたとすると、そのペアは c に分類される。Jaccard 係数はこのような分類を行った後、次の式で計算される。

$$Jaccard\ 係数 = \frac{a}{a+b+c}$$

なお、他の外的基準を用いた評価手法として、Jaccard 係数によく似た Rand 係数がある。Rand 係数は、Jaccard 係数では使用しなかった d のセルを使い、次の式で計算される。

$$Rand\ 係数 = \frac{a+d}{a+b+c+d}$$

現在の研究では、Jaccard 係数と Rand 係数の相関関係は 0.937 と非常に高く、Jaccard 係数が 0.32 の標準偏差を持つのに対し、Rand 係数が 0.20 の標準偏差を持つことが分かっている [5]。そのため、本論文では Jaccard 係数のみを評価基準として採用した。

表 3: Circle のクラスタリング結果

	Jaccard 係数	平均クラスタ数	処理時間 (msec)
K-means 法	0.581	3.0	22.0
SOON1	0.664	3.3	438.8
RCON	1.0	3.0	928.2

5.2 評価結果

5.2.1 人工データセットでの評価

RCON の性能を、人工データセット、実データセット二種類のクラスタリング結果から比較する。クラスタリングの性能を比較するアルゴリズムは、K-means 法、SOON1[1]、RCON の 3 種類である。評価は各データセットに対し、それぞれの手法を 10 回ずつ行い、得られる Jaccard 係数の平均を計測する。また、評価の際の各手法のパラメータの設定は、K-means 法については正解セットのクラスタ数を、他の手法は Jaccard 係数が最大になるように手動で設定する。

本実験で使用した人工データセットはそれぞれ特徴をもつ 2 種類の 2 次元データセットである。表 2 に各データセットの概要を示す。Circle データセットは、クラスタに属するデータの数がそれぞれ異なる円形のデータセットである。Amoeba データセットは [6] で使用された、データが円状に分布しておらずクラスタが入り組んでいるデータセットである。

Circle データセットのクラスタリング結果は表 3、及び図 7 の通りである。Jaccard 係数を比較すると、RCON が他の 2 つの手法に比べ、良い結果を得ている。平均クラスタ数は各手法ともほぼ同じだが、処理時間で見ると K-means 法が最も早く、RCON が最も遅い結果となる。分布図を比較してみると、K-means 法、SOON1 では各クラスタに間違って分類したデータが見られるが、RCON では正解通りに分割できている。

Amoeba データセットのクラスタリング結果は表 4、図 8 で示した結果となる。Jaccard 係数を見てみると、Circle データセットの場合と比べてみても、RCON が K-means 法、SOON1 に比べ非常に高い数値を示している。平均クラスタ

表 4: Amoeba のクラスタリング結果

	Jaccard 係数	平均クラスタ数	処理時間 (msec)
K-means 法	0.315	4.0	37.7
SOON1	0.331	3.2	156.5
RCON	1.0	4.0	315.5

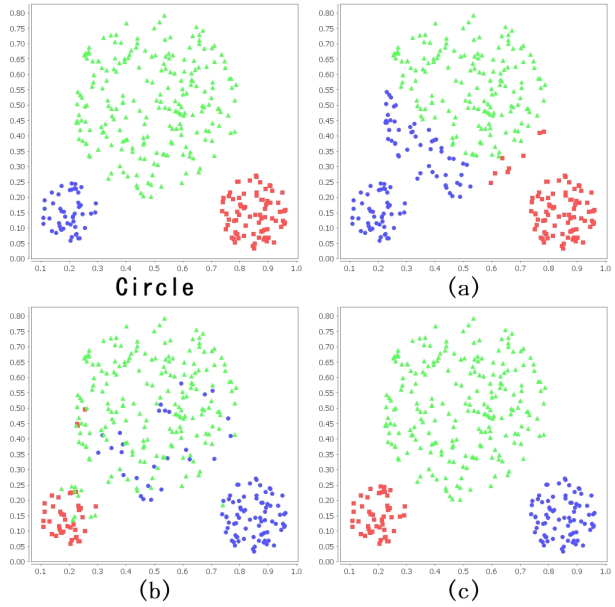

図 7: Circle のデータ分布及びクラスタリング結果 ((a)K-means 法,(b)SOON1,(c)RCON)

数と処理時間については、Circle データセットの場合とほぼ同じ結果となった。図 8 の分布図を見てみると、K-means 法と SOON1 では不定形のデータ分布に関係なくクラスタリングされている。一方、RCON では不定形のデータ分布に従ってクラスタを形成している。

5.2.2 実データセットでの評価

UCI Machine Learning Repository[7] に収録されたデータセットの中で、特徴的な結果を示した Haberman's Survival データセットについて結果を説明する。Haberman's Survival データセットはデータ数 306 個、3 次元の要素を持つ、2 つのクラスタからなるデータセットである。Haberman データセットの各手法でのクラスタリング結果を表 5、図 9 に示す。なお、図 9 は、データセットに主成分分析を行い、その第一主成分と第二主成分をプロットしたものである。Jaccard 係数を比較してみると、K-means 法と RCON が比較的よい結果を得られたのに対し、SOON1 は低い数値となる。ただし、K-means 法は予め正解セットのクラスタ数を与えられているので、K-means 法の Jaccard 係数の高さがそのまま性能の高さにならない可能性がある。また、平均クラスタ数でみると SOON1 が 4.4 なのに対し、RCON は 18.0 と他の手法に比べ細かいクラスタに分割された。処理時間は人工データセットのときと同様、K-means 法が最も早く、RCON が最も遅い結果となる。図 9 で見てみると、K-means 法は手法の特性上、概ね正しくクラスタリングされ、SOON1 は中央のクラスタの分割が上手くできていないことが確認できる。RCON では左側のデータが密集したクラスタの分割は概ね正しいが、データの分布密度が比較的低い中央のクラスタが細かく分割されている。

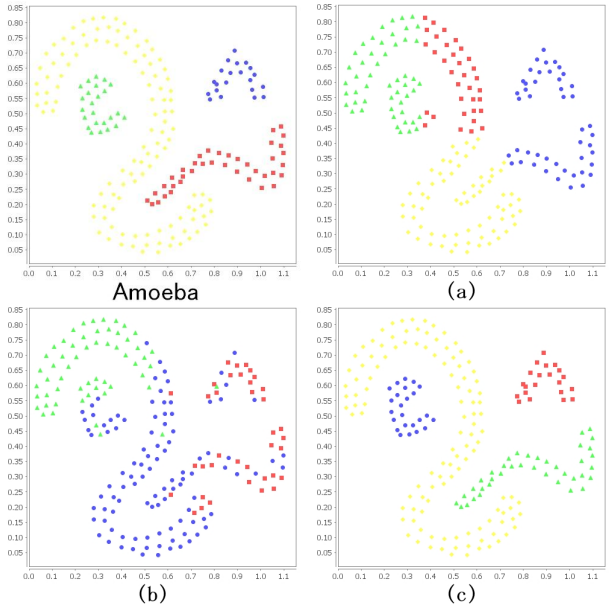

図 8: Amoeba のデータ分布及びクラスタリング結果 ((a)K-means 法,(b)SOON1,(c)RCON)

表 5: Haberman のクラスタリング結果

	Jaccard 係数	平均クラスタ数	処理時間 (msec)
K-means 法	0.889	2.0	23.4
SOON1	0.624	4.4	312.5
RCON	0.807	18.0	713.9

6 考察

各データセットでの評価結果から RCON の性能を考察する。Circle データセットの結果は、各アルゴリズムの性質の違いをよく表している。K-means 法は、クラスタの重心を考えているため、各クラスタの要素数が同じ程度でなければ良いクラスタリング結果を得られない。SOON1 もアルゴリズムの性質上、分割がほぼ同じ大きさになるため、クラスタの大きさが違うと望む結果を得られない。一方、RCON では予め、相対近傍グラフによりデータを関係付けているため、クラスタの要素数や大きさが異なっても分類出来る。

Amoeba データセットでは、各クラスタリング手法の任意形状のクラスタの分割能力が見て取れる。K-means 法、SOON1 は共にクラスタの形状が円状または楕円状であることを仮定しており、Amoeba データセットのようにクラスタが複雑に入り組んでいる状況を想定していない。しかし、RCON では相対近傍グラフにより、近隣のデータとの関係を考慮しているため、クラスタの形状に依存せずに分類出来る。

Haberman データセットでの比較結果は、RCON の弱点を示唆している。Haberman データセットのように各クラスタのデータ分布密度に差がある場合、クラスタリング結果からわかるように、RCON では適切なクラスタ数で分割されない可能性がある。この弱点を克服するためには、[6] の手法に見られるようなデータの密度や、階層的手法のクラスタの統合を考慮したアルゴリズムが必要である。

最後に RCON の処理時間について考察する。RCON と、その基となった SOON1 の処理時間の差は、データ数が多くなるにつれて指数的に大きくなる。RCON と SOON1 の自

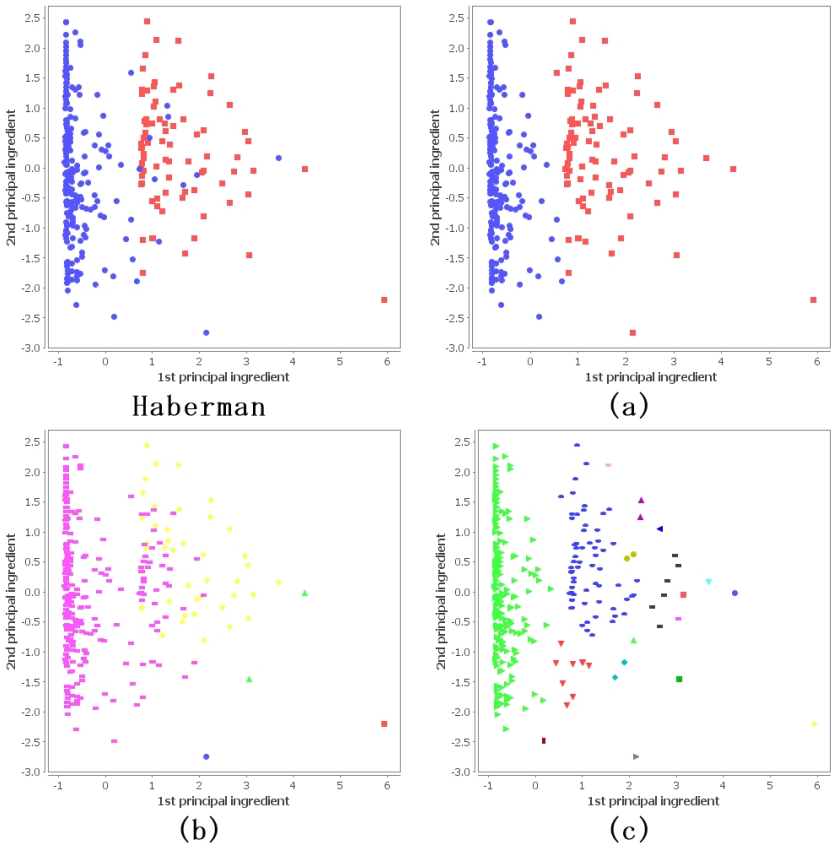

図 9: Haberman のデータ分布及びクラスタリング結果 ((a)K-means 法,(b)SOON1,(c)RCON)

己組織化手法はほぼ同じことから、相対近傍グラフの作成に時間が掛かると考えられる。相対近傍グラフ作成部の現在のアルゴリズムでは $O(n^3)$ のコストだが、改良を行うことにより、その計算コストを $O(n \log n)$ にまで減らせるという報告がある [8]。よって、今後の改良によっては処理時間の短縮を期待できる。

7 結論

パルス結合振動子の自己組織化に際し、事前に相対近傍グラフによるデータ間の関係付けを行うことにより、任意形状のクラスタを分割可能なクラスタリングアルゴリズムを開発できた。しかしながら、クラスタのデータ分布に差があるデータセットに対しては、適切なクラスタ数で分割されない可能性があり、この課題を解決する各種手法を試していきたい。また、処理時間は現在の実装において指数オーダーだが、今後の改良によって処理時間の短縮を図りたい。

謝辞: 本研究は総務省戦略的情報通信研究開発制度 (SCOPE) の委託事業によるものである。

参考文献

[1] M. B. H. Rhouma and H. Frigui, “Self-organization of pulse-coupled oscillators with application to clustering,” IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, pp.180-195, (2001)

[2] R. E. Mirollo and S. H. Strogatz, “Synchronization of pulse-coupled biological oscillators,” Society for Industrial and Applied Mathematics Journal on Applied Mathematics, vol. 50, pp. 1645-1662, (1990)

[3] Yoshiaki Taniguchi, Naoki Wakamiya and Masayuki Murata, “A traveling wave based communication mechanism for wireless sensor networks,” Journal of Networks, vol. 2, pp. 24-32, (2007)

[4] R.Urquhart,“Graph Theoretical Clustering Based on Limited Neghbourhood Sets,” Pattern Recognition, Vol. 15, No. 3, pp. 173-187, (1982)

[5] G. W. Milligan,“A monte carlo study of thirty internal criterion measures for cluster analysis,” Psychometrika, vol.46, pp.187-199, (1981)

[6] M.Ester, H.-P.Kriegel, J.Sander, and X.Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” KDD1996, pp.226-231, (1996)

[7] UCI Machine Learning Repository : http://archive.ics.uci.edu/ml/

[8] 杉原厚吉,“なわばりの数理モデルーボロノイ図からの数理工学入門ー,” 共立出版, (2009)