研究速報

コードベクトルの分離・結合を考慮した自己組織化マップに基づくクラスタリング手法

A Clustering Method based on a Self-Organizing Map Showing the Separation and Combination of Code Vectors

正会員　今村弘樹[†]，　藤村　誠[††]，正会員　黒田英夫[†††]

Hiroki Imamura[†], Makoto Fujimura[††] and Hideo Kuroda[†††]

Abstract　A method is proposed for clustering data that results in more accurate clustering when clusters are close to each other. The method takes into account the continuity of the data by using a self-organizing map showing the separation and combination of code vectors. Using this map reduces the dependence of fitting of code vectors. The proposed method can be thus used to obtain accurate clustering.

キーワード：クラスタリング，自己組織化マップ，データの連続性，コードベクトルの分離・結合

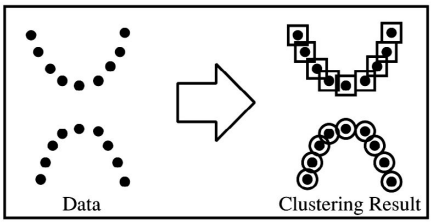


図 1　データ間の距離に基づく手法によるクラスタリング例#1
An example #1 of clustering by the method based on distance between data.

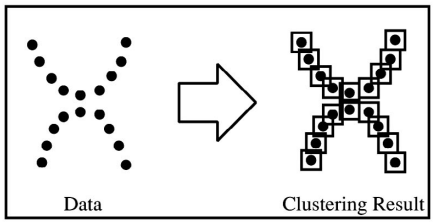


図 2　データ間の距離に基づく手法によるクラスタリング例#2
An example #2 of clustering by the method based on distance between data.

1. ま え が き

クラスタリング手法の主な手法として，k-means 法[1)〜5)]，fuzzy-c 平均法[6)〜10)]がある．これらの手法は，クラスタリングするデータの分布が正規分布状である場合，高精度にクラスタリングが可能である．しかし，クラスタリングするデータの分布が任意の形状である場合，クラスタリング精度が著しく低下する場合がある．そこで，データの分布が任意の形状である場合においても高精度にクラスタリングを行うために，任意形状クラスタリング法が提案された[11)12)]．これらの手法は，同じクラスタに含まれるデータ間の距離は，異なるデータ間の距離より短いという前提に基づきクラスタリングを行っている（**図 1**）．このため，異なるクラスタが接近している場合やクラスタリングするデータ数が少ない場合，異なるクラスタが同じクラスタとしてクラスタリングされてしまうことがある（**図 2**）．

そこで筆者らは，自己組織化マップ (SOM) に基づくクラスタリングする手法を提案した[13)]．この手法では，まず，代表ベクトルをランダムに生成し，それらに順番を付ける．次に，それらを順番に連結したコードベクトルを設定する（**図 3**）．ただし，図 3 の□は，代表ベクトルを表し，代表ベクトル間の直線は，それぞれの連結線を表す．さらに，SOM のアルゴリズムに基づきコードベクトルをデータに収束（フィッティング）させる．フィッティングの後，コードベクトルをつなぐ角度が大きく変化する毎にコードベク

2008 年 8 月 16 日受付，2009 年 4 月 6 日最終受付，2009 年 4 月 17 日採録

†創価大学 工学部 情報システム工学科
（〒 192–8577 東京都八王子市丹木町 1–236, TEL 042-691-9400）
††長崎大学 工学部 情報システム工学科
（〒 852–8521 長崎市文教町 1–14, TEL 095-819-2574）
††長崎大学 大学院　生産科学研究科
（〒 852–8521 長崎市文教町 1–14, TEL 095-819-2574）
†School of Information System, Department of Engineering, Soka University
(1–236, Tangi-cho, Hachioji City, Tokyo, 192–8577)
††Dept. of Computer and Information Sciences, Nagasaki University
(1–14, Bunkyou-mach, Nagasaki City, 852–8521)
††Graduate School of Science and Technorogy, Nagasaki University
(1–14, Bunkyou-mach, Nagasaki City, 852–8521)

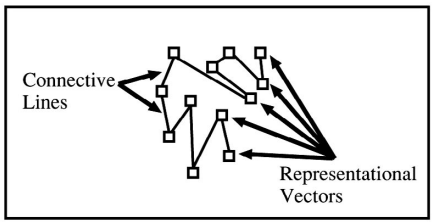


図 3　コードベクトルの構成
The construction of the code vector.

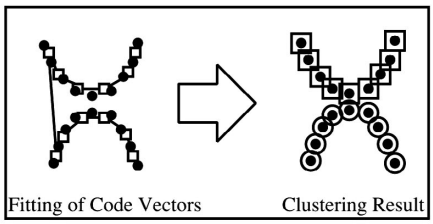


図 4　SOM に基づく手法によるクラスタリング例#1
An example #1 of clustering by the method based on SOM.

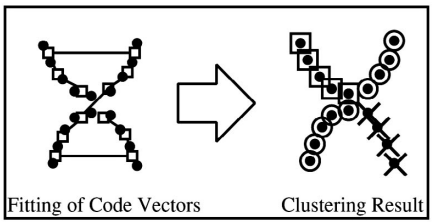


図 5　SOM に基づく手法によるクラスタリング例#2
An example #2 of clustering by the method based on SOM.

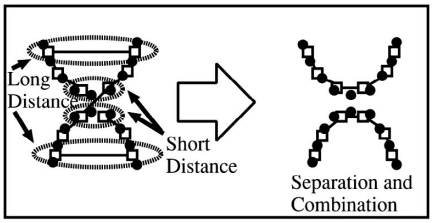


図 6　距離によるコードベクトルの連結と分離
Separation and combination of code vectors according to distance.

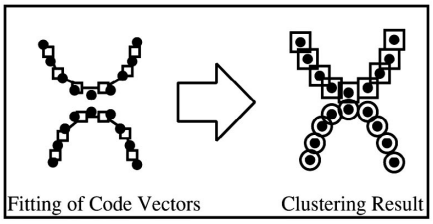


図 7　SOM に基づく手法によるクラスタリング例#3
An example #3 of clustering by the method based on SOM.

トルを分け，ラベリングを行う．次に，それぞれのデータが最小距離となるコートベクトルを探索し，そのコードベクトルのラベルをデータに割り当てることにより，各データをクラスタリングしている (図 4)．しかし，コードベクトルの初期値はランダムに設定されるため，初期値の状況に応じてフィッティングの状況も変化し，本来，異なるクラスタであるデータが同じクラスタのデータとしてクラスタリングされたり，本来，同じクラスタであるデータが異なるクラスタのデータとしてクラスタリングされる場合がある (図 5)．そこで，例えば，図 5 のようにコードベクトルが収束する場合において，コードベクトルをデータにフィッティングさせる際に，コードベクトル間の距離が長いコードベクトルは連結を分離し，コードベクトル間の距離が短いコードベクトルは連結を結合する．この処理により，コードベクトルが各クラスタ毎に良好にフィッティングできると考える (図 6)．さらに，各々連結したコードベクトルに対してラベリングを行い，各コードベクトルのラベルをそのコードベクトルに最近傍となるデータに割り当てることによりクラスタリングを行う．これにより，コードベクトルの初期値の依存性が低くなり，良好なクラスタリング結果が得られると考える (図 7)．そこで本研究では，SOM の各コードベクトルをコードベクトル間の距離の閾値により，分離・結合する SOM に基づくクラスタリング手法を提案する．

2.　従来手法と提案手法のアプローチ

ここでは，まず，データ間の距離に基づく任意形状クラスタリング，SOM に基づく従来のクラスタリング，それぞれのアプローチを示し，次に，それらに対する提案手法のアプローチを示すことにより，提案手法の有効性を示す．なお，以下で示す予備実験とは，クラスタリングが良好に行われるようにパラメータを人手で調整する実験を表す．

2.1　データ間の距離に基づく任意形状クラスタリングのアプローチ

ここでは，データ間の距離に基づく任意形状クラスタリング手法の中で比較的クラスタリング精度の高い文献 12) の手法のアプローチを示す．

クラスタリングするデータ $\boldsymbol{x}_i$ を i 番目の k 次元ベクトルとし，全データ数を n として，t=1 とする．

$$D^{(t)} = \arg \min_{1 \leq j \leq n, 1 \leq k \leq n, j \neq k} ||\boldsymbol{x}_j - \boldsymbol{x}_k|| \qquad (1)$$

を算出し，$D^{(t)}$ が予備実験により決定する閾値 $Th_{(i)}$ 以内であれば，$\boldsymbol{x}_j$，$\boldsymbol{x}_k$ を抽出し，同じラベル値を与える．$\boldsymbol{x}_j$，$\boldsymbol{x}_k$ が異なるラベル値であった場合，ラベル値の大きなデータのラベル値と同じラベル値を持つデータすべてのラベルをラベル値の小さなデータのラベル値に置き換える．さらに，$D^{(t+1)}(D^{(t+1)} > D^{(t)})$ となる $\boldsymbol{x}_j$，$\boldsymbol{x}_k$ を抽出し，同様の処理を行う．$Th_{(i)}$ を超えない範囲で，t をインクリメントし，この処理を繰り返すことにより，ラベル付けを行う．最後にラベル値毎にクラスタリングを行う．

この手法は，データ間の距離の短いデータから順番にラベリングを行っているために，図 1 に示すような異なるクラスタが離れている場合は，高精度にクラスタリングが可

能である．しかし，図 2 に示すような異なるクラスタが近接している場合，また，クラスタリングするデータ数が少ない場合，同じクラスタとして，クラスタリングされる恐れがある．

2.2 SOM に基づく従来のクラスタリングのアプローチ

ここでは，SOM に基づく従来のクラスタリング手法[13]のアプローチを示す．

まず，k 次元のベクトルである各データ $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_v$ にフィッティングさせる k 次元のコードベクトルの初期値を設定する．ただし，コードベクトルは先端 $\boldsymbol{m}_1$ から末端 $\boldsymbol{m}_w$ へ順番に番号が付けられているものとする．次に，SOM に基づく繰り返し計算により，コードベクトルをクラスタリングするデータにフィッティングさせていく．各データとその最近傍のコードベクトルの最大距離が閾値以内となった場合，フィッティングを終了する．ここで，コードベクトルの先端 $(i=1)$ から末端 $(i=n)$ に対して，

$$\frac{(\boldsymbol{m}_{i+2}-\boldsymbol{m}_{i+1})^T(\boldsymbol{m}_{i+1}-\boldsymbol{m}_i)}{||\boldsymbol{m}_{i+2}-\boldsymbol{m}_{i+1}||||\boldsymbol{m}_{i+1}-\boldsymbol{m}_i||}<Th_{s1} \qquad (2)$$

かつ，

$$||\boldsymbol{m}_{i+2}-\boldsymbol{m}_{i+1}||-||\boldsymbol{m}_{i+1}-\boldsymbol{m}_i||>0 \qquad (3)$$

が成り立つ i が存在するならば，$\boldsymbol{m}_{i+2}$ 以降のコードベクトルのラベル値は，$\boldsymbol{m}_i$，$\boldsymbol{m}_{i+1}$ と異なるラベル値を与える．ただし，式 (2) の左辺は，ベクトル $(\boldsymbol{m}_{i+2}-\boldsymbol{m}_{i+1})$ と $(\boldsymbol{m}_{i+1}-\boldsymbol{m}_i)$ のなす角における cos の値を表し，Th_{s1} は，式 (2) における閾値を表す．なお，Th_{s1} は，予備実験により決定する．$\boldsymbol{m}_{i+2}$ が $\boldsymbol{m}_{i+1}$ と $\boldsymbol{m}_i$ と異なるクラスタであれば，$\boldsymbol{m}_{i+2}$ と $\boldsymbol{m}_{i+1}$，$\boldsymbol{m}_{i+1}$ と $\boldsymbol{m}_i$ の間が不連続となり，式 (2) で，左辺が閾値以下となる．また，その際に式 (3) に示すように，$\boldsymbol{m}_{i+2}$ と $\boldsymbol{m}_{i+1}$ 間の距離が $\boldsymbol{m}_{i+1}$ と $\boldsymbol{m}_i$ 間の距離よりも大きくなる．最終的に，各データに対して最近傍となるコードベクトルのラベル値を各データに割り当て，全データをクラスタリングする．

以上の様に，まず，コードベクトルをデータにフィッティングさせ，コードベクトルが連続性を持つ部分毎にラベル付けし，最終的に，各データに対して最近傍となるコードベクトルのラベル値を各データに割り当て，全データをクラスタリングすることにより，異なるクラスタが近接している場合，また，クラスタリングするデータ数が少ない場合でも，良好にクラスタリングが行える（図 4）．ただし，コードベクトルの最終的なフィッティング状態がコードベクトルの初期値に大きく依存するため，クラスタリング結果は，コードベクトルの初期状態に大きく依存するという課題がある（図 5）．

2.3 提案手法のアプローチ

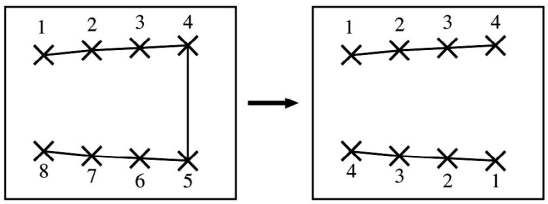
ここでは，提案手法のアプローチを示す．提案手法も，文献 12) と同じく，コードベクトルを SOM のアルゴリズムに基づき，データにフィッティングさせることにより，クラスタリングを行う．ただし，コードベクトルをデータにフィッティング際に，コードベクトルを分離と結合させることにより，従来手法の SOM に基づく手法で課題となっていたコードベクトルの初期値に対する依存性を低減させる．提案手法では，まず，k 次元のベクトルである各データ $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_v$ にフィッティングさせる k 次元のコードベクトルの初期値を設定する．ただし，コードベクトルは先端 $\boldsymbol{m}_1$ から末端 $\boldsymbol{m}_w$ へ順番に番号が付けられているものとする．次に，SOM に基づく繰り返し計算により，コードベクトルをクラスタリングするデータにフィッティングさせていく．ここで，ある閾値 Th_{s2} を予備実験により設定し，コードベクトルの先端 $(i=1)$ から末端 $(i=w)$ に対して，

$$||\boldsymbol{m}_{i+1}-\boldsymbol{m}_i||>Th_{s2} \qquad (4)$$

を満たす i において，$i+1$ 以降のコードベクトルは，i 以前のコードベクトルと異なる連結を持つコードベクトルとして分離する (**図 8**)．この際，$i+1$ 以降のコードベクトルの順番は，先端から末端まで，順番を新たに付け直すこととする．また，

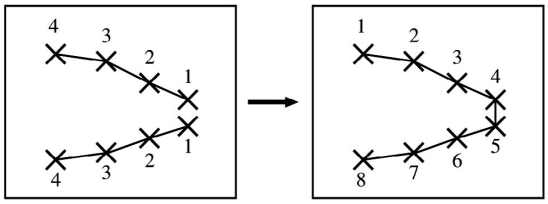
$$||\boldsymbol{m}_e^{\alpha}-\boldsymbol{m}_{e'}^{\beta}||\leqq Th_{s2} \qquad (5)$$

を満たすコードベクトルの連結を結合させる (**図 9**)．ただし，α，β は，それぞれ異なるコートベクトルの連結を表し，e，e' はそれぞれ，α，β における先端，または，末端の番号を表す．なお，連結後は，先端から末端まで順番を付け直すこととする．上記の処理を各データとその最近傍のコードベクトルの最大距離が閾値以内となるまで繰り返し，各連結毎に異なるラベルを割り当てる．最終的に，各データに対して最近傍となるコードベクトルのラベル値を各データに割り当て，全データをクラスタリングする．



Separation

図 8 コードベクトルの分離
Separation of code vectors.



Combination

図 9 コードベクトルの連結
Combination of code vectors.

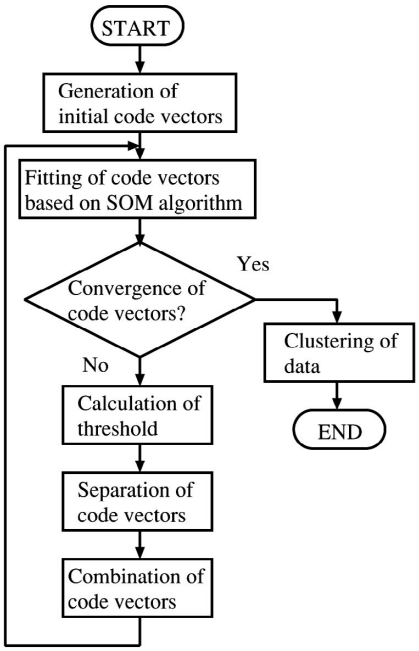


図 10 提案手法の処理の流れ図
Flow of the process in the proposed method.

3. 提案手法のアルゴリズム

ここでは，提案手法のアルゴリズムを示す．**図 10** に提案手法のアルゴリズムの流れ図を示す．なお，提案手法では，1 次の SOM を用いることとする．また，以下に示すアルゴリズムの 1. においてコードベクトルの生成とラベル，フラグの初期化，2. から 4. において SOM に基づくコードベクトルのフィッティング，5. においてコードベクトルの分離と結合で用いる閾値の計算，6. においてコードベクトルの分離処理，7. においてコードベクトルの結合処理，8. において各データにおける最近傍のコードベクトルのラベル値をそのデータに割り当て，クラスタリングを行っている．

1. クラスタリングするデータは k 次元のベクトル $\boldsymbol{x}$ とし，データ数は n 個とする．まず，k 次元のコードベクトル $\boldsymbol{m}_p^l$ を定義域内においてランダムに生成する．ただし，$\boldsymbol{m}_p^l$ は，コードベクトルのラベル $l(1 \leqq l \leqq Cl)(Cl$ はラベルの最大値を表す定数) における $p(1 \leqq p \leqq num_l)(num_l$ はラベル l におけるコードベクトルの番号の最大値を表す定数) 番目におけるコードベクトルを表す．初期値として，$C_l = 1$，すべてのコードベクトルのラベルを 1 とし，コードベクトルのラベル 1 における先端のコードベクトルのフラグを $e_1^1 = 1$，コードベクトルのラベル 1 における末端のコードベクトルのフラグを $e_{num_1}^1 = 2$，コードベクトルのラベル 1 における上記以外のコードベクトルのフラグを $e_a^1 = 0(a = 2, 3, \cdots, num_1 - 1)$ とする．

2. 以下を $t = 1, 2, \cdots; j = 1, 2, \cdots n$ について繰り返す．

3. 次の式

$$(l^*, p^*) = \arg \min_{1 \leqq l \leqq C_l, 1 \leqq p \leqq num_l} ||\boldsymbol{m}_p^l - \boldsymbol{x}_j|| \qquad (6)$$

を満たす $\boldsymbol{m}_{p^*}^{l^*}$ を j 番目のデータ $\boldsymbol{x}_j$ に対する勝利コードベクトルと定義する．つまり，勝利コードベクトルは，$\boldsymbol{x}_j$ に対して最近傍のコードベクトルを表す．

4. 勝利コードベクトルとその周辺のコードベクトルを

$$\boldsymbol{m}_p^l = \begin{cases} \boldsymbol{m}_p^l + \gamma(t)\{\boldsymbol{x}_j - \boldsymbol{m}_p^l\} \\ \text{if } (p \in p^* + N_c) \wedge (l = l^*) \\ \\ \boldsymbol{m}_p^l \\ \text{if } (p \notin p^* + N_c) \vee (l \neq l^*) \end{cases} \qquad (7)$$

により更新する．ただし，N_c は，p^* に対する N_c 近傍を表す．また，

$$\gamma(t) = \frac{0.7}{1 + [t/7]} \qquad (8)$$

とし，$[t/7]$ は，$t/7$ を超えない最大の整数を表す．なお，式 (8) は，予備実験により決定した．ここで，以下の式

$$\arg \max_{1 \leqq l \leqq C_l, 1 \leqq p \leqq num_l} ||\boldsymbol{m}_p^l - \boldsymbol{x}_j|| < Th_3 \qquad (9)$$

を満たす場合，8. へ．ただし，Th_3 は，式 (9) における閾値を表す．なお，Th_3 は，予備実験により決定する．

5. $l_1 = 1, 2, \cdots, C_l; p = 1, 2, \cdots, num_{l_1}$ について，5.1. を繰り返す．ただし，num_{l_1} は，l_1 に含まれるコードベクトルの数とする．また，$d_{total} = 0$，$d_{cnt} = 0$ とする．

5.1. もし，$\boldsymbol{m}_{p+1}^{l_1}$ が存在するなら，

$$d_{diff} = ||\boldsymbol{m}_{p+1}^{l_1} - \boldsymbol{m}_p^{l_1}|| \qquad (10)$$

を算出し，$d_{total} = d_{total} + d_{diff}$，$d_{cnt} = d_{cnt} + 1$ とする．

5.2. $d_{cnt} \neq 0$ ならば，$Th = (d_{total}/d_{cnt}) + Th_\psi d_{diff}^{(max)}$ とする．ただし，$d_{diff}^{(max)}$ は，d_{diff} の最大値とし，Th_ψ は，$d_{diff}^{(max)}$ に対する重み係数とする．Th_ψ は，予備実験により決定する．

6. $l_1 = 1, 2, \cdots, C_l; p = 1, 2, \cdots, num_{l_1}$ について，6.1. を繰り返す．

6.1. もし，$\boldsymbol{m}_{p+1}^{l_1}$ が存在し，

$$||\boldsymbol{m}_{p+1}^{l_1} - \boldsymbol{m}_p^{l_1}|| \geqq Th_{s2} \qquad (11)$$

を満たすなら，$e_p^{l_1} = 2$，$e_{p+1}^{l_1} = 1$ とし，$l_1 + 1$ から C_l に含まれるコードベクトルすべてのラベルを一つインクリメントする．さらに，l_1 に含まれる $p + 1$ から num_{l_1} までのコードベクトルのラベルを一つインクリメントし，

$\boldsymbol{m}_\phi^{l_1+1}(\text{after}) := \boldsymbol{m}_{p+\phi}^{l_1}(\text{before})$
for $\forall \phi \in \{1, 2, \cdots, num_{l_1+1}\}$

とする．なお，ここでは，分離・結合前後のコードベクトルを区別するため，それぞれを (before)，(after) という記述で表す．また，:= は，右辺を左辺に代入することを表す．**図 11** に p=4，$p + 1$=5 とした場合におけるアルゴリズム 6.1. の実行例を示す．

7. $l_1 = 1,2,\cdots,C_l; l_2 = l_1+1, l_1+2,\cdots,C_l$; $p = 1,2,\cdots,num_{l_1}; q = 1,2,\cdots,num_{l_2}$ について，以下を繰り返す．ただし，num_{l_1},num_{l_2} はそれぞれ，l_1,l_2 に含まれるコードベクトルの数とする．繰り返し処理終了後，2. へ．

7.1. もし，$e_p^{l_1}$=1，または，2，かつ，$e_q^{l_2}$=1，または，2 ならば，

$$d = ||\boldsymbol{m}_p^{l_1} - \boldsymbol{m}_q^{l_2}|| \tag{12}$$

を算出し，$d < Th_{s2}$ ならば，以下の処理を行う．

7.1.1.もし，$e_p^{l_1}$=1，かつ，$e_q^{l_2}$=1 ならば，

(1) $\boldsymbol{m}_\phi^{l_1}$(after):=$\boldsymbol{m}_{num_{l_2}-\phi+1}^{l_2}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_2}\}$

(2) $\boldsymbol{m}_{num_{l_2}+\phi}^{l_1}$(after):=$\boldsymbol{m}_\phi^{l_1}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_1}\}$

(3) $e_1^{l_1} := 1, e_{num_{l_1}+num_{l_2}}^{l_1} := 2$

(4) $e_\phi^{l_1} := 0$ for $\forall\phi\in\{2,3,\cdots,num_{l_1}+num_{l_2}-1\}$

とすることにより l_1，l_2 に含まれるコードベクトルの結合を行い，7. へ．**図 12** に num_{l_1}=4，num_{l_2}=4 の場合におけるアルゴリズム 7.1.1. の実行例を示す．

7.1.2.もし，$e_p^{l_1}$=2，かつ，$e_q^{l_2}$=2 ならば，

(1) $\boldsymbol{m}_\phi^{l_1}$(after):=$\boldsymbol{m}_\phi^{l_2}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_2}\}$

(2) $\boldsymbol{m}_{num_{l_2}+\phi}^{l_1}$(after):=$\boldsymbol{m}_{num_{l_1}-\phi+1}^{l_1}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_1}\}$

(3) $e_1^{l_1} := 1, e_{num_{l_1}+num_{l_2}}^{l_1} := 2$

(4) $e_\phi^{l_1} := 0$ for $\forall\phi\in\{2,3,\cdots,num_{l_1}+num_{l_2}-1\}$

とすることにより l_1，l_2 に含まれるコードベクトルの結合を行い，7. へ．**図 13** に num_{l_1}=4，num_{l_2}=4 の場合におけるアルゴリズム 7.1.2. の実行例を示す．

7.1.3.もし，$e_p^{l_1}$=1，かつ，$e_q^{l_2}$=2 ならば，

(1) $\boldsymbol{m}_\phi^{l_1}$(after):=$\boldsymbol{m}_\phi^{l_2}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_2}\}$

(2) $\boldsymbol{m}_{num_{l_2}+\phi}^{l_1}$(after):=$\boldsymbol{m}_\phi^{l_1}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_1}\}$

(3) $e_1^{l_1} := 1, e_{num_{l_1}+num_{l_2}}^{l_1} := 2$

(4) $e_\phi^{l_1} := 0$ for $\forall\phi\in\{2,3,\cdots,num_{l_1}+num_{l_2}-1\}$

とすることにより l_1，l_2 に含まれるコードベクトルの結合を行い，7. へ．**図 14** に num_{l_1}=4，num_{l_2}=4 の場合におけるアルゴリズム 7.1.3. の実行例を示す．

7.1.4.もし，$e_p^{l_1}$=2，かつ，$e_q^{l_2}$=1 ならば，

(1) $\boldsymbol{m}_{num_{l_1}+\phi}^{l_1}$(after):=$\boldsymbol{m}_\phi^{l_2}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_2}\}$

(2) $\boldsymbol{m}_\phi^{l_1}$(after):=$\boldsymbol{m}_\phi^{l_1}$(before)
for $\forall\phi\in\{1,2,\cdots,num_{l_1}\}$

(3) $e_1^{l_1} := 1, e_{num_{l_1}+num_{l_2}}^{l_1} := 2$

(4) $e_\phi^{l_1} := 0$ for $\forall\phi\in\{2,3,\cdots,num_{l_1}+num_{l_2}-1\}$

とすることにより l_1，l_2 に含まれるコードベクトルの結合を行い，7. へ．**図 15** に num_{l_1}=4，num_{l_2}=4 の場合におけるアルゴリズム 7.1.4. の実行例を示す．

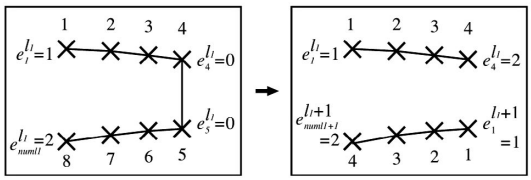


図 11 アルゴリズム 6.1. の例
An example of algorithm 6.1.

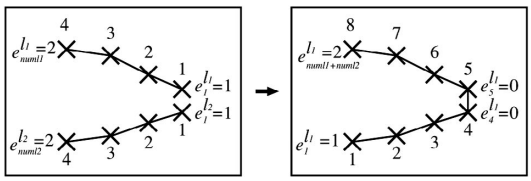


図 12 アルゴリズム 7.1.1. の例
An example of algorithm 7.1.1.

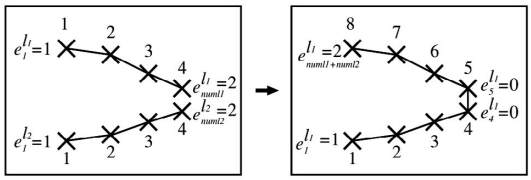


図 13 アルゴリズム 7.1.2. の例
An example of algorithm 7.1.2.

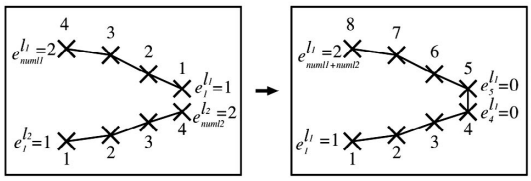


図 14 アルゴリズム 7.1.3. の例
An example of algorithm 7.1.3.

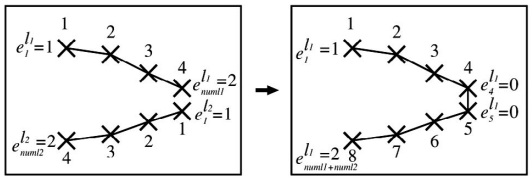


図 15 アルゴリズム 7.1.4. の例
An example of algorithm 7.1.4.

8. 各 $\boldsymbol{x}_i(i = 1,2,\cdots,n)$ において，

$$(l', p') = \arg \min_{1\leqq l\leqq C_l, 1\leqq p\leqq num_l} ||\boldsymbol{x}_i - \boldsymbol{m}_p^l|| \tag{13}$$

となる $\boldsymbol{m}_{p'}^{l'}$ におけるコードベクトルのラベル l をそれぞれ割り当て，すべての処理を終了する．

4. 実　験

正規分布以外の形状データに対する提案手法の有効性を評価するために，コンピュータにより生成した人工データに対して，クラスタリングを行った．ここでは，文献 13) の手法を従来手法とし，この従来手法，文献 12) の手法と提案手法との比較を行った．なお，文献 12) の手法のパラメータは，σ=20.0 とし，従来手法のパラメータは，

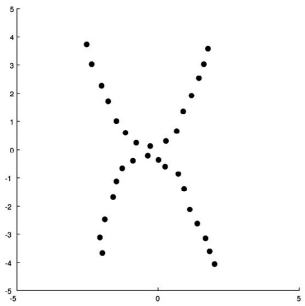


図 16 生成した人工データ#1
Generated synthesis data#1.

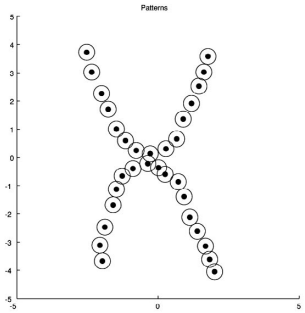


図 17 クラスタリング結果 (文献 12) の手法)
The result of clustering (the method in literature[12]).

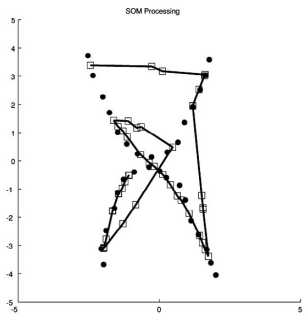


図 18 繰り返し計算終了後におけるコードベクトルの状態 (従来手法)
The state of code vectors after final repetition calculation(the conventional method).

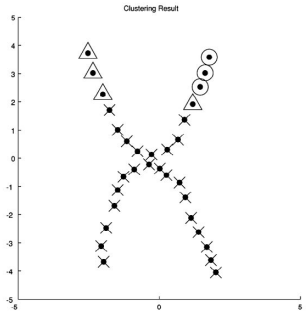


図 19 クラスタリング結果 (従来手法)
The result of clustering(the conventional method).

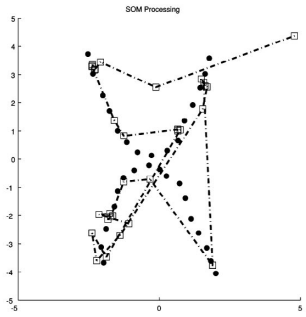


図 20 繰り返し計算 2 回目におけるコードベクトルの状態 (提案手法)
A state of code vectors in the 2nd repetition calculation(the proposed method).

コードベクトル数は 40，Nc=3，SOM 処理時の収束判定の閾値 Th_1=5.0×10^{-3}，Th_2=cos(π/2)，Th_3=1.0 とし，提案手法のパラメータは，コードベクトル数は 40，Nc=3，Th_ψ = 8.5×10^{-1}，Th_{s2}=1.0 とした．なお，ここで用いたパラメータの値は，各手法のクラスタリング精度が最大となるように予備実験により決定した．

まず，異なるデータが近接している際の提案手法の有効性を示すために，**図 16** に示す空間的な連続性を持つ二つのクラスタが存在するデータに対して実験を行った．**図 17** は，文献 12) の手法によるクラスタリング結果を示している．ここでは，空間的な連続性を持つ二つのクラスタを抽出したいのだが，この手法では，データ全体が一つのクラスタとして抽出されてしまった．**図 18** は，従来手法における繰り返し計算終了後におけるコードベクトルの状態を示している．コードベクトルがデータに良好にフィッティングできていないことがわかる．**図 19** は，従来手法におけるクラスタリング結果を示している．コードベクトルがデータに良好にフィッティングできていないことにより，クラスタリング結果も良好な結果が得られなかった．また，**図 20** から**図 22** は，提案手法における，それぞれの繰り返し計算過程におけるコードベクトルの状態を示している．提案手法では，コードベクトルが分離と結合を行い，クラスタの形状にフィットしてゆく過程がわかる．**図 23** は，提案手法におけるクラスタリング結果を示している．クラスタリング結果から提案手法は，文献 12) の手法や従来手法に比べ，良好にクラスタリングが行われていることがわかる．

次に，前の実験に対してクラスタ数とクラスタの形状を変化させた際の提案手法の有効性を評価するために，**図 24** に示す空間的な連続性を持つ三つのクラスタが存在するデータに対して実験を行った．**図 25** は，文献 12) の手法によるクラスタリング結果を示している．ここでは，空間的な連続性を持つ三つのクラスタを抽出したいのだが，この手法では，データ全体が一つのクラスタとして抽出されてしまった．**図 26** は，従来手法における繰り返し計算終了後におけるコードベクトルの状態を示している．コードベクトルがデータに良好にフィッティングできていないことがわかる．**図 27** は，従来手法におけるクラスタリング結果を示している．コードベクトルがデータに良好にフィッティングできていないことにより，クラスタリング結果も良好な結果が得られなかった．また，**図 28** から**図 30** は，提案手法における，それぞれの繰り返し計算過程におけるコー

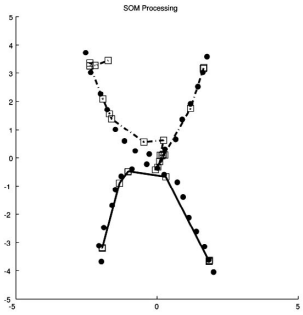


図 21 繰り返し計算 3 回目におけるコードベクトルの状態 (提案手法)
A state of code vectors in the 3rd repetition calculation(the proposed method).

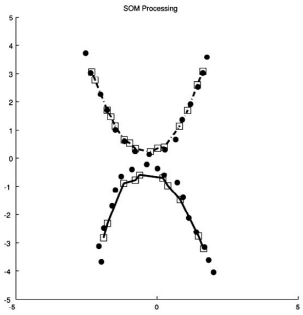


図 22 繰り返し計算終了後におけるコードベクトルの状態 (提案手法)
The state of code vectors after final repetition calculation(The proposed method).

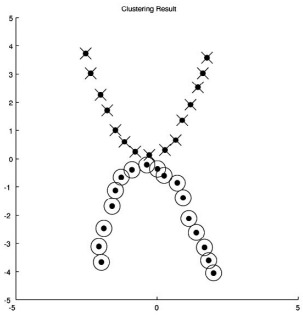


図 23 クラスタリング結果 (提案手法)
The result of clustering(the proposed method).

ドベクトルの状態を示している．提案手法では，コードベクトルが分離と結合を行い，各クラスタの中央にフィッティングしていく過程がわかる．**図 31** は，提案手法におけるクラスタリング結果を示している．クラスタリング結果から提案手法は，文献 12) の手法や従来手法に比べ，良好にクラスタリングが行われていることがわかる．以上の結果は，フィッティング精度を高めるために，提案手法では，コードベクトルを分離・結合させデータにフィッティングさせることにより，フィッティング精度が向上したことによる効果であると考える．ただし，異なるクラスタが交差するような場合は，交差する箇所でコードベクトルが連結し，本来，異なるクラスタが同じクラスタとしてクラスタリングされる可能性が高いので，提案手法を適用するのは難しいと考える．

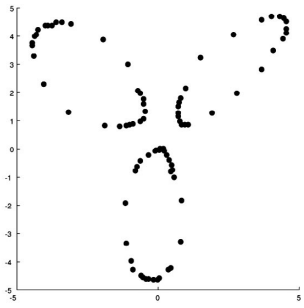


図 24 生成した人工データ#2
Generated synthesis data#2.

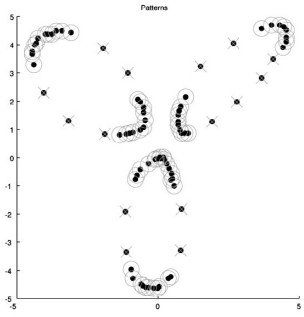


図 25 クラスタリング結果 (文献 12) の手法)
The result of clustering (the method in literature[12]).

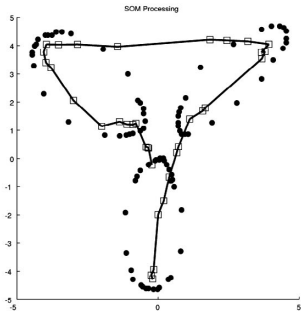


図 26 繰り返し計算終了後におけるコードベクトルの状態 (従来手法)
The state of code vectors after final repetition calculation(the conventional method).

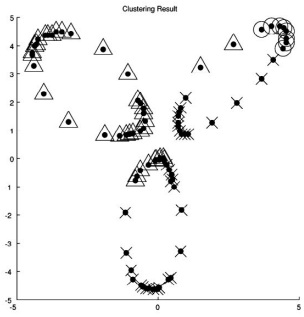


図 27 クラスタリング結果 (従来手法)
The result of clustering(the conventional method).

5. むすび

フィッティングの状況によっては，本来，異なるクラスタであるデータが同じクラスタのデータとしてクラスタリングされたり，本来，同じクラスタであるデータが，異な

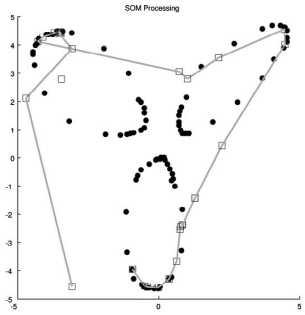


図 28 繰り返し計算 2 回目におけるコードベクトルの状態（提案手法）
A state of code vectors in the 2nd repetition calculation(the proposed method).

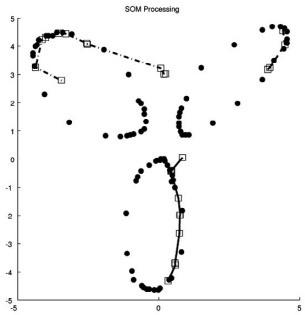


図 29 繰り返し計算 9 回目におけるコードベクトルの状態（提案手法）
A state of code vectors in the 3rd repetition calculation(the proposed method).

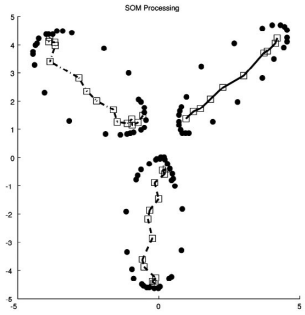


図 30 繰り返し計算終了後におけるコードベクトルの状態（提案手法）
The state of code vectors after final repetition calculation(The proposed method).

るクラスタのデータとしてクラスタリングされる場合がある従来の SOM に基づくクラスタリング手法に対して，本研究では，SOM の各コードベクトルをコードベクトル間の距離の閾値により，分離したり，結合する SOM に基づくクラスタリング手法を提案した．人工データに対する実験より，文献 12) の手法，従来手法に対して，提案手法は良好なクラスタリング結果を得た．今後は，提案手法を実データのクラスタリングに適用することにより，その有効性を評価する予定である．

〔文　献〕

1）B. T. Cover and P. Hart: “Nearest Neighbor Classification,” IEEE Trans. on Information Theory, **IT–13**, 1, pp.21–27(1967)
2）K. Fukunaga: “Introduction to statistical pattern recognition,” Academic press, Boston, 2 edition(1990)
3）R. O. Duda, P. E. Hart and D. G. Stocrk: “Pattern Clasification – Second Edition,” Wiley Interscience(2002)
4）井上 光平, 浦浜 喜一: “次元削減に基づくフィルタリングによる kNN 識別の高速化,” 信学論, **J85–D–II**, 5, pp.950–953(2002)
5）春日 秀雄, 山本 博章, 岡本 正行: “高速 K-means 法を用いたカラー画像の色量子化,” 信学論, **J82–D–II**, 7, pp.1120–1128(1999)
6）R. L. Cannon, J. V. Dave, and J. C. Bezdek: “Efficient implementation of the fuzzy c-means clustering algorithms,” IEEE Trans. on PAMI, **8**, 2, pp.248–255(1986)
7）R. E. Hammah and J. H. Curran: “Validity Measures for the Fuzzy Cluster Analysis of Orientations,” IEEE Trans. on PAMI, **22**, 12, pp.1467–1472(2000)
8）A. Keller and F. Klawonn: “Fuzzy Clustering with Weighting of Data Variables,” Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, **8**, 6, pp.735–746(2000)
9）高橋 正人, 服部 和雄: “ファジー c-means 法と最近傍決定則を用いたクリスプなクラスタリング法,” 信学論, **J83–D–II**, 9, pp.1957–1961(2000)
10）井上 光平, 浦浜 喜一: “緩和反復法に基づくロバストファジークラスタリング,” 信学論, **J85–D–II**, 6, pp.1140–1143(2002)
11）井上光平, 浦浜喜一: “データ間連結度に基づく任意形状ファジークラスタの抽出,” 信学論, **J86-D-II**, 10, pp.1511-1513(2003)
12）今村 弘樹, 藤村 誠, 黒田 英夫: “クラスタ間距離の昇順によるラベリングに基づくノイズにロバストな任意形状クラスタリング,” 映情学会誌, **60**, 4, pp.618-620(2006)
13）今村 弘樹, 藤村 誠, 黒田 英夫: “自己組織化マップに基づくデータの連続性を考慮したクラスタリング手法,” 映情学会誌, **60**, 8, pp.1312-1316(2006)
14）A.S.Georghiades, P.N. Belhumeur and J.D.Kriegman: “From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose,” IEEE Trans. on PAMI, **23**, 6 pp.643-660(2001)

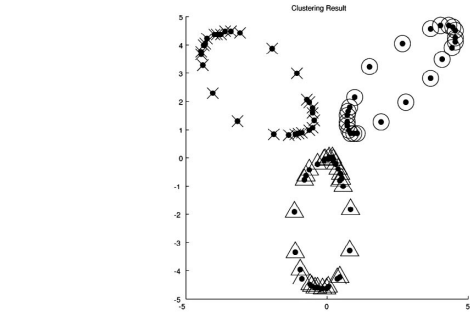


図 31 クラスタリング結果 (提案手法)
The result of clustering(the proposed method).

今村　弘樹（いまむら　ひろき） 1997 年，創価大学工学部情報システム学科卒業．2002 年，米国カーネギーメロン大学ロボティクス研究所訪問研究員．2003 年，北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．同年，長崎大学工学部情報システム工学科助手，2007 年，同助教，2009 年，創価大学工学部情報システム工学科准教授，現在に至る．博士（情報科学）．画像処理，パターン認識，コンピュータグラフィックスの研究に従事．正会員

藤村　誠（ふじむら　まこと） 1985 年，福井大学工学部卒業．同年，FHL に入社．1990 年，長崎大学工学部助手，1994 年，同講師，2007 年，同准教授，現在に至る．動画像の高性能符号化，画像処理などの研究に従事．

黒田　英夫（くろだ　ひでお） 1971 年，九州工業大学大学院修士課程修了．同年，日本電信電話公社電気通信研究所に入社．1989 年，長崎大学工学部大学院教授．その間，1994 年，シドニー大学客員教授，現在に至る．工学博士．画像信号高能率符号化，画像処理，CG，CV 等の研究に従事．正会員．