数理解析研究所講究録 1365 巻 2004 年 1-9

On the convergence in self-organization and its applications
(自己組織化における収束性とその応用について)

秋田県立大学 システム科学技術学部　星野満博 (Mitsuhiro Hoshino)
秋田県立大学 システム科学技術学部　木村 寛 (Yutaka Kimura)
Faculty of Systems Science and Technology, Akita Prefectural University

1. 自己組織化マップ

本報告は，以下に示す自己組織化マップ（self-organizing maps）と呼ばれるアルゴリズムにおける諸問題についての一考察である．

自己組織化マップのモデルは，Kohonen 等による研究 [5] が有名であるが，ここでは，ノード，ノードの値，入力，学習の 4 要素により特徴づけることとする．

(I) E をノードの空間とし，距離空間であるものとする．ノードの集合 I は E のある部分集合であるものとする．
(II) 各ノードは，その値をもつ．A をノードの値の空間とし，ノルム空間であるものとする．ノード i をその値 $a(i)$ に対応させる写像 $a : I \to A$ をリファレンス関数と呼ぶ．R をリファレンス関数全体の集合とする．$R = \{a \mid a : I \to A\}$.
(III) 入力集合 X は A のある部分集合であるものとする．$x \in X$ を入力と呼ぶ．
(IV) 入力 x とリファレンス関数 a が与えられとき，リファレンス関数 a は，学習関数 $L_{a,x} : I \to [0,1]$ によって定まる学習率 $L_{a,x}(i)$ に従い入力 x を学習する．この結果，リファレンス関数 a は次のように定義されるリファレンス関数 a' へと更新される．

$$\begin{aligned} a'(i) &= (1 - L_{a,x}(i))a(i) + L_{a,x}(i)x \\ &= a(i) + L_{a,x}(i)(x - a(i)). \end{aligned} \tag{1.1}$$

今，初期リファレンス関数 a_0 と入力列として $x_0, x_1, x_2, \ldots \in X$ が与えられたとする．このとき，

$$\begin{aligned} a_{k+1}(i) &= (1 - L_{a_k,x_k}(i))a_k(i) + L_{a_k,x_k}(i)x_k \\ &= a_k(i) + L_{a_k,x_k}(i)(x_k - a_k(i)) \end{aligned} \tag{1.2}$$

によってリファレンス関数 $a_1, a_2, a_3 \ldots$ が逐次に生成される．この生成過程または，X と生成されたリファレンス関数との対応を自己組織化マップと呼ぶことにする．

一般に，適用する問題に応じて，想定するノードの空間，ノード値の空間，および学習方法等，様々なバリエーションが考えられる．通常，応用上はノード集合 I として，$\boldsymbol{Z}$（整数全体）または $\boldsymbol{Z}^2$ の有限部分集合と対等な集合が用いられているが，距離の入れ方は様々である．

本報告においては，学習方法として等式 (1.1) および (1.2) を採用する．学習関数 $L_{a,x}$ として，次のように定義されるタイプのものを考える．各リファレンス関数 $a \in R$，入力 $x \in X$ に対して，集合 $I(a,x)$ を

$$I(a,x) = \left\{ i^* \in I \,\middle|\, \|a_k(i^*) - x_k\| = \inf_{i \in I} \|a_k(i) - x_k\| \right\} \tag{1.3}$$

によって定義する．また，写像 $J : R \times X \to I$ は $J(a,x) \in I(a,x)$ を満たし，関数 $L : [0,\infty) \to [0,1]$ は単調減少であるものとする．このとき，学習関数として，次のタイプのものがある．

(i) 学習関数 $L^1_{a,x} : I \to [0,1]$ を

$$L^1_{a,x}(i) = L\Big(\inf_{j \in I(a,x)} d_E(j,i) \Big) \tag{1.4}$$

によって定義する．ただし，$d_E(\cdot,\cdot)$ は $E \times E$ 上で定義される距離関数を表わす．

(ii) 学習関数 $L^2_{a,x} : I \to [0,1]$ を

$$L^2_{a,x}(i) = L\Big(d_E(J(a,x),i) \Big) \tag{1.5}$$

によって定義する．

2. 基本的な例

ここでは，自己組織化マップの基本的な例として，以下の例を考える．

ノード値 $\mathbb{R}$，1 次元ノード配列の場合:

(1) ノード集合 $I = \{1,2,\ldots,n\} \subset \mathbb{N}$.
(2) ノードの値の空間を $\mathbb{R}$（実数値全体，ユークリッドノルム）とし，初期リファレンス関数を $a_0 : I \to \mathbb{R}$ とする．
(3) 入力列 $x_0, x_1, x_2, \ldots \in X \subset \mathbb{R}$.
(4) 学習方法

$$J(a,x) = \min I(a,x), \quad a \in R, x \in X,$$

$$N_\varepsilon(i) = \Big\{ j \in I \,\Big|\, |j-i| \le \varepsilon \Big\} : \quad i \text{の近傍},$$

$0 \leq \alpha \leq 1$： 学習率,

$$L_{a,x}(i) = \begin{cases} \alpha & i \in N_\varepsilon(J(a,x)) \\ a_k(i) & i \notin N_\varepsilon(J(a,x)) \end{cases} \quad : \text{ 学習関数,}$$

$$a_{k+1}(i) = (1 - L_{a_k,x_k}(i))a_k(i) + L_{a_k,x_k}(i)x_k, \quad k = 0, 1, 2, \ldots .$$

つまり，今 n 個のノード $1, 2, \ldots, n$ があり，そのそれぞれに対してノードの値 $a_0(1), a_0(2), \ldots, a_0(n)$ が与えられている．このとき，入力とこれに伴う学習により各ノードの値が更新される．$x_0 \in X$ が入力されたならば，$a_0(1), a_0(2), \ldots, a_0(n)$ のなかで x_0 と最も近いものを選び，その値に対応するノード i^* とその周囲のノード i に対して学習

$$a_1(i) = (1 - \alpha)a_1(i) + \alpha x_1$$

が適用され，それ以外のノードに対しては学習が適用されず，$a_1(i) = a_0(i)$ となる．入力 $x_1, x_2, x_3, \ldots$ に対して，これを繰り返すことにより，逐次にノードの更新がおこなわれ，同時にリファレンス関数 $a_1, a_2, a_3, \ldots$ が逐次に生成される．

上述のような1次元ノード配列でノードの値が実数値の場合，ノードとノードの値の対応を表すリファレンス関数 $a : I \to \mathbb{R}$ を

$$a = [a(1), a(2), \ldots, a(n)] \tag{2.1}$$

と表すことにする．

例 1 ノード集合を $I = \{1, 2, 3, 4, 5\}$，ノードの値の空間を $\mathbb{R}$，初期リファレンス関数を $a_0 = [2, 3, 1, 5, 4]$ とし，入力列を $4, 5, 2, 1, 3, 4, 3, 5, 4, 1, 5, \ldots$ とする．学習方法は次のように定める．

$$J(a_k, x_k) = \min I(a_k, x_k) = \min \left\{ i^* \in I \,\middle|\, |a_k(i^*) - x_k| = \inf_{i \in I} |a_k(i) - x_k| \right\},$$

$$\begin{aligned} N_1(J(a_k, x_k)) &= \left\{ j \in I \,\middle|\, |j - J(a_k, x_k)| \leq 1 \right\} \\ &= \{J(a_k, x_k) - 1, J(a_k, x_k), J(a_k, x_k) + 1\} \cap I : \quad \text{学習範囲,} \end{aligned}$$

$$\alpha = \frac{1}{2} : \quad \text{学習率,}$$

$$a_{k+1}(i) = \begin{cases} \frac{1}{2}a_k(i) + \frac{1}{2}x_k & i \in N_1(J(a_k, x_k)), \\ a_k(i) & i \notin N_1(J(a_k, x_k)). \end{cases} \quad k = 0, 1, 2, \ldots .$$

このとき，ノードの値は次のように更新される．
初期リファレンス関数 a_0 と入力 $x_0 = 4$ から

$$J(a_0, x_0) = 5, N_1(5) = \{4, 5\},$$
$$a_1(1) = a_0(1) = 2, \quad a_1(2) = a_0(2) = 3, \quad a_1(3) = a_0(3) = 1,$$
$$a_1(4) = \frac{a_0(4)+x_0}{2} = 4.5, \quad a_1(5) = \frac{a_0(5)+x_0}{2} = 4$$
$$a_1 = [2,\ 3,\ 1,\ 4.5,\ 4].$$

リファレンス関数 a_1 と入力 $x_1 = 5$ から

$$J(a_1, x_1) = 4, \quad N_1(4) = \{3, 4, 5\},$$
$$a_2(1) = a_1(1) = 2, \quad a_2(2) = a_1(2) = 3, \quad a_2(3) = \frac{a_1(3)+x_1}{2} = 3,$$
$$a_2(4) = \frac{a_1(4)+x_1}{2} = 4.75, \quad a_2(5) = \frac{a_1(5)+x_1}{2} = 4.5,$$
$$a_2 = [2,\ 3,\ 3,\ 4.75,\ 4.5].$$

これを繰り返すことにより

$$a_3 = [2,\ 2.5,\ 3,\ 4.75,\ 4.5],$$
$$a_4 = [1.5,\ 1.75,\ 3.,\ 4.75,\ 4.5],$$
$$a_5 = [1.5,\ 2.375,\ 3.,\ 3.875,\ 4.5],$$
$$a_6 = [1.5,\ 2.375,\ 3.5,\ 3.9375,\ 4.25],$$
$$a_7 = [1.5,\ 2.6875,\ 3.25,\ 3.46875,\ 4.25],$$
$$a_8 = [1.5,\ 2.6875,\ 3.25,\ 4.23438,\ 4.625],$$
$$a_9 = [1.5,\ 2.6875,\ 3.625,\ 4.11719,\ 4.3125],$$
$$a_{10} = [1.25,\ 1.84375,\ 3.625,\ 4.11719,\ 4.3125].$$
$$\cdots\cdots$$

が得られる． □

上の例1において，初期リファレンスおよび入力列は特別意味のある数，数列ではないが

$$k \geq 5 \text{ のとき } \quad a_k(1) \leq a_k(2) \leq a_k(3) \leq a_k(4) \leq a_k(5)$$

が成り立っている．このように，学習を繰り返すことにより，リファレンス関数が単調性等の性質をもち，各ノードの値の配列にある種の規則性が現れることがあるが，上記の過程において現れるこのような現象は組織化とよばれている．実際，様々なノード集合，ノードの値の空間，学習方法において，このような組織化現象を含む幾つかの興味深い現象が現れる．また，これらの性質を利用することにより，多

くの実問題への応用が成されている．しかしながら，組織化等の現象は常に起こるわけではなく，実アプリケーションにおいては，求めるべき結果を得る為に，モデルに応じた学習方法を採用している．ここでの，どのように学習をさせるべきかという問題は非常に難解であると言われている．また，一般に巧く学習させると，組織化された状態，すなわちノードの値が整列した状態に収束する．別の言い方をするならば，要求する良い状態へ収束するように学習させる．例1のように，学習率が固定された値の場合は，いつまでも学習が繰り返され収束しないが，多くの応用において，具体的には，徐々に学習率を小さくして，最終的に0にすることにより強制的に収束させている．

3. 1次元ノード配列の場合における基本性質について

例1においては，各ステップでのノードの値の配列は，初期ノードの値と，入力列に依存することがわかる．このようなモデルにおいて学習率が一定の場合，各ステップでのノードの値への影響を考えるとき，初期ノードの値の依存度と初期の入力値の依存度は徐々に薄れてくる．そこで，入力を入力集合 X の値をとる確率変数の実現値として扱うことにする．

以下では，前節で定義した実数のノード値をとる1次元ノード配列の場合に限定する．x を確率空間 $(S,\mathcal{B},P)$ 上で定義された $X=\mathbb{R}$ の値をとる確率変数であるとする．

今，入力条件と学習方法に対して次のような条件を定義する．

入力条件 1 入力変数 x に対して，$P(\{x=c\})>0$ を満たす $c\in\mathbb{R}$ が存在しない．

学習方法 1 学習方法を次のように定義する．

$$\text{学習範囲:}\quad N_1(J(a_k,x_k))=\left\{j\in I\;\middle|\;|j-J(a_k,x_k)|\le 1\right\}$$

$$\text{ただし } J(a_k,x_k)=\min\left\{i^*\in I\;\middle|\;|a_k(i^*)-x_k|=\inf_{i\in I}|a_k(i)-x_k|\right\},$$

$$\text{学習率:}\quad 0\le\alpha\le 1,$$

$$\text{学習:}\quad a_{k+1}(i)=\begin{cases}(1-\alpha)a_k(i)+\alpha x_k & i\in N_1(J(a_k,x_k)),\\ a_k(i) & i\notin N_1(J(a_k,x_k)).\end{cases}\qquad k=0,1,2,\ldots.$$

上記の設定のもとで以下の基本的な性質が成り立つ．

定理 1 学習方法1を仮定するとき，リファレンス関数 a_k について

(1) a_k が I 上で単調増加であるならば a_{k+1} も I 上で単調増加である．
(2) a_k が I 上で単調減少であるならば a_{k+1} も I 上で単調減少である．

定理 2　入力条件1および学習方法1を仮定するとき，あるkに対して，a_kはI上で単調となる (a.e.).

これらの性質を利用した応用として，巡回セースルマン問題がある．巡回セールスマン問題においては，通常，ノードの値の空間として，上記の設定である1次元ユークリッド空間ではないが，2次元のユークリッド空間を用いる．また，入力集合として巡回する都市全体を考え，更に巡回を考慮させる為にノードの空間に1次元配列の両端を結合したループ状の位相構造を導入する．

定理 3　入力条件1および学習方法1を仮定するとき，期待値の極限

$$\lim_{k\to\infty} E\Big[\frac{1}{k+1}\sum_{l=0}^{k} a_l(i)\Big],\quad i\in I$$

が存在する．

定理 4　入力条件1および学習方法1を仮定する．$w(i)$を

$$w(i)=\lim_{k\to\infty} E\Big[\frac{1}{k+1}\sum_{l=0}^{k} a_l(i)\Big],\quad i\in I$$

によって定義する．このとき

(1) wはI上で単調である．
(2) $z(i)$を

$$z(i)=\begin{cases} w(i) & (w\text{が}I\text{上で単調増加のとき}) \\ w(n+1-i) & (w\text{が}I\text{上で単調減少とき}) \end{cases},\quad i\in I$$

によって定義するとき$z(1),z(2),\ldots,z(n)$は次の等式を満たす．

$$\int_{g(i)}^{h(i)} (x-z(i))P(dx)=0,\quad i\in I$$

ただし

$$h(i)=\begin{cases} \frac{z(i+1)+z(i+2)}{2} & (i=1,2,\cdots,n-2), \\ \infty & (i=n-1,n), \end{cases}$$

$$g(i)=\begin{cases} \frac{z(i-2)+z(i-1)}{2} & (i=3,4,\cdots,n), \\ -\infty & (i=1,2). \end{cases}$$

例 2　定理3,4における期待値 $w(i), i = 1, 2, \ldots, n$ を求めると次のようになる.

x が $(0, 1]$ 上の一様分布に従う場合

ノードの個数	期待値 w
$n = 1$	$w = [\frac{1}{2}]$
$n = 2$	$w = [\frac{1}{2},\ \frac{1}{2}]$
$n = 3$	$w = [\frac{3}{10},\ \frac{5}{10},\ \frac{7}{10}]$ または $w = [\frac{7}{10},\ \frac{5}{10},\ \frac{3}{10}]$
$n = 4$	$w = [\frac{5}{20},\ \frac{7}{20},\ \frac{13}{20},\ \frac{15}{20}]$ または $w = [\frac{15}{20},\ \frac{13}{20},\ \frac{7}{20},\ \frac{5}{20}]$
$n = 5$	$w = [\frac{2}{10},\ \frac{3}{10},\ \frac{5}{10},\ \frac{7}{10},\ \frac{8}{10}]$ または $w = [\frac{8}{10},\ \frac{7}{10},\ \frac{5}{10},\ \frac{3}{10},\ \frac{2}{10}]$

x が標準正規分布 $N(0, 1)$ に従う場合

ノードの個数	期待値 w
$n = 1$	$w = [0]$
$n = 2$	$w = [0,\ 0]$
$n = 3$	$w \fallingdotseq [-0.613426,\ 0,\ 0.613426]$
	または $w \fallingdotseq [0.613426,\ 0,\ -0.613426]$
$n = 4$	$w \fallingdotseq \left[-\sqrt{\frac{2}{\pi}},\ -0.44852,\ 0.44852,\ \sqrt{\frac{2}{\pi}}\right]$
	または $w \fallingdotseq \left[\sqrt{\frac{2}{\pi}},\ 0.44852,\ -0.44852,\ -\sqrt{\frac{2}{\pi}}\right]$
$n = 5$	$w \fallingdotseq [-1.00285,\ -0.613426,\ 0,\ 0.613426,\ 1.00285]$
	または $w \fallingdotseq [1.00285,\ 0.613426,\ 0,\ -0.613426,\ -1.00285]$

ただし $w = [w(1), w(2), \ldots, w(n)]$ である. □

4.　2次元ノード配列の場合について

ここでは，ノードの配列が2次元で，ノードの値が実数である場合について考える．1次元の場合と同様に以下のように定義することができる.

(1) ノード集合 $I = \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$. 距離として $d(i, j) = |i - j|$ を用いる.

(2) ノードの値の空間を $\mathbb{R}$（実数値全体，ユークリッドノルム）とし，初期リファレンス関数を $a_0 : I \to \mathbb{R}$ とする．リファレンス関数 $a : I \to \mathbb{R}$ を

$$a = \begin{bmatrix} a(1,1) & a(1,2) & \ldots & a(1,n) \\ a(2,1) & a(2,2) & \ldots & a(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ a(m,1) & a(m,2) & \ldots & a(m,n) \end{bmatrix}$$

と記すことにする.

(3) 入力変 x は数 $\mathbb{R}$ の値をとる確率変数であるとする．$x_0, x_1, x_2, \ldots$ を x の実現値の列とする.

(4) 学習方法

学習範囲: $N_\varepsilon(J(a_k,x_k))=\left\{(i,j)\in I \;\middle|\; d((i,j),J(a_k,x_k))\le\varepsilon\right\}$,

ただし

$$J(a_k,x_k)=\min\left\{(i^*,j^*)\in I \;\middle|\; |a_k(i^*,j^*)-x_k|=\min_{(i,j)\in I}|a_k(i,j)-x_k|\right\},$$

ここでの I における順序として例えば辞書式順序を用いる.

学習率: $0\le\alpha\le 1$.

学習: $$a_{k+1}(i,j)=\begin{cases}(1-\alpha)a_k(i,j)+\alpha x_k & (i,j)\in N_\varepsilon(J(a_k,x_k)),\\ a_k(i,j) & (i,j)\notin N_\varepsilon(J(a_k,x_k)),\end{cases}$$
$$k=0,1,2,\ldots.$$

ノード配列が 2 次元の場合，次の意味において 1 次元ノード配列の場合に成立した単調性の保存が成り立たなくなる.

例えば，d をユークリッド距離として，学習範囲を

$$N(J(a_k,x_k))=\left\{(i,j)\in I \;\middle|\; d((i,j),J(a_k,x_k))\le\sqrt{2}\right\}$$

とするとき，各成分を固定した場合，すべての $a_k(i,\cdot),a_k(\cdot,j)$ が（単調な数列という意味において）単調であっても，$a_{k+1}(i,\cdot)$ および $a_{k+1}(\cdot,j)$ は単調であるとは限らない.

例 3　上記のような学習範囲と学習率として $\frac{1}{2}$ を用いる場合を考える．ここで，I における順序として辞書式順序を用いることにする.

$$a_0=\begin{bmatrix}1&2&7&8\\2&3&8&9\\3&4&9&10\\4&5&10&11\end{bmatrix}$$

であるとする．このとき，すべての $a_0(i,\cdot),a_0(\cdot,j)$ は単調増加である．ここで，入力値として $x_k=4$ であったならば，学習によりノードの値は

$$a_{k+1}=\begin{bmatrix}1&2&7&8\\3&3.5&6&9\\3.5&4&6.5&10\\4&4.5&7&11\end{bmatrix}$$

と更新され，関数 $a_{k+1}(\cdot,3)$ は単調ではない. □

参考文献

[1] P. Billingsley, *Probability and Measure*, John Wiley & Sons, 1979.

[2] R. M. Dudley, *Real Analysis and Probability*, Wadsworth & Brooks, 1989.

[3] E. Erwin, K. Obermayer and K. Schulten, *Self-organization maps: stationary states, metastability and convergence rate*, Bio. Cybern., 67 (1992), pp. 35–45

[4] E. Erwin, K. Obermayer and K. Schulten, *Self-organization maps: ordering, convergence properties and energy functions*, Bio. Cybern., 67 (1992), pp. 47–55

[5] T. Kohonen, *Self-Organizing Maps, Third Edition*, Springer, 2001.

[6] P. L. Zador, *Asymptotic quantization error of continuous signals and the quantization dimension*, IEEE Trans. Inform. Theory, vol. IT-28, No. 2, March (1982), pp. 139–149.