Competing and Accommodating Behaviors of Peace SOM

Haruna MATSUSHITA* and Yoshifumi NISHIO*
* Department of Electrical and Electronic Engineering, Tokushima University
2–1 Minami-Josanjima, Tokushima 770–8506, Japan
Telephone: +81–88–656–7470, Fax: +81–88–656–7471, Email: {haruna, nishio}@ee.tokushima-u.ac.jp

***Abstract*— The Self-Organizing Map (SOM) attracts attentions for clustering in these years. In our past study, we have proposed a method using simultaneously two kinds of SOMs whose features are different, namely, one self-organizes the area on which input data are concentrated, and the other self-organizes the whole of the input space. Further, we have applied this method to clustering of data including a lot of noises and have confirmed the efficiency. However, in order to obtain an efficient clustering performance using this method, we must determine the appropriate number of the SOMs used in the method. In this study, we propose the Peace SOM (PSOM) algorithm which possesses both competing and accommodating abilities. The competing and the accommodating behaviors of PSOM are investigated with applications to clustering input data including a lot of noises. We can see that PSOM successfully extracts clusters even in the case that we do not know the number of clusters in advance.**

I. Introduction

Since we can accumulate a huge amount of data including useless information, it is important to investigate various extraction methods of clusters from data including a lot of noises. The Self-Organizing Map (SOM) attracts attentions for clustering in these years. SOM is an unsupervised neural network introduced by Kohonen in 1982 [1] and is a model simplifying self-organization process of the brain. SOM can obtain a statistical feature of input data and can be applied to a wide field of data classifications [2]-[5].

In our past study, we have proposed a method using simultaneously two kinds of SOMs whose features are different (nSOM method) [6]. In the nSOM method, one self-organizes the area on which input data are concentrated, and the other self-organizes the whole of the input space. We have investigated competing behavior of nSOM caused by the difference of the initial states and the neighborhood functions. Furthermore, we have applied nSOM to clustering of data including a lot of noises and have confirmed the efficiency. However, in order to obtain an efficient clustering performance using nSOM method, we must determine the appropriate number of the SOMs used in the method, namely, we must know the number of clusters in advance. If we use too many SOMs in this method, some SOMs compete with each other in the same cluster and the one cluster is extracted as wrong plural clusters.

In this study, we propose the Peace SOM (PSOM) algorithm which possesses both competing and accommodating abilities. We use this algorithm after executing the nSOM method. This algorithm makes the competing SOMs in the same cluster to accommodate the cluster — hence named "peace", while the SOMs situated far keep to compete with other SOMs.

The differences between nSOM and PSOM are as follows.
1) The number of winner neurons for one input data.
In nSOM, only one winner neuron is selected from all the neurons in the all SOMs. In PSOM, one winner neuron is selected from every SOM. Namely, k winner neurons are selected from k SOMs.
2) The number of updated SOMs for one input data.
In nSOM, only one SOM including the winner neuron is updated. In PSOM, some SOMs whose winner neuron is near the input data are updated.

In the Section II, the algorithm of the two kinds of SOMs (nSOM) is introduced. In the Section III, we explain the learning algorithm of the proposed PSOM algorithm in detail. The competing and the accommodating behaviors of PSOM are investigated in the Section IV with applications to clustering input data including a lot of noises. For 2-dimensional input data, extraction ability of clusters is evaluated both visually and quantitatively using a correct answer rate. We also apply PSOM to 5-dimensional input data and confirm the extraction ability for higher-dimensional data. We can see that PSOM successfully extracts clusters even in the case that we do not know the number of clusters in advance.

II. Two kinds of SOMs (nSOM)

In our past study, we have proposed a method using simultaneously two kinds of SOMs whose features are different (nSOM method), namely, one self-organizes the area on which input data are concentrated, and the other self-organizes the whole of the input space. We call the former $\mathrm{SOM_L}$ and the latter $\mathrm{SOM_G}$. In order to apply nSOM to clustering, we use totally n SOMs, that is one $\mathrm{SOM_G}$ and $(n-1)$ $\mathrm{SOM_L}$; namely $\mathrm{SOM_{L1}}, \mathrm{SOM_{L2}}, \cdots, \mathrm{SOM_{L}}_{(n-1)}$. We use d-dimensional input data $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$ including a lot of noises.

The differences between $\mathrm{SOM_G}$ and $\mathrm{SOM_L}$ are only the initial states and the neighborhood functions. However, these differences cause interesting competing behaviors of nSOM. The difference between the initial states of $\mathrm{SOM_G}$ and $\mathrm{SOM_L}$ is as follows. The initial values of all the weight vectors $\boldsymbol{w}_{Ll}$ of $\mathrm{SOM_L}_l$ $(l = 1, 2, \cdots, n-1)$ are given in the overall of the input space at random. The initial values of all the weight

vectors $\boldsymbol{w}_G$ of $\mathrm{SOM_G}$ are given around the center of the input space at random. The second difference is the neighborhood function of each SOM. The learning rate of $\mathrm{SOM_L}$ is given as decreasing rapidly, on the other side, the learning rate of $\mathrm{SOM_G}$ decreases monotonically with time. In addition, the width of the neighborhood function of $\mathrm{SOM_G}$ is given as decreasing slowly, however the width of the neighborhood function of $\mathrm{SOM_L}$ decreases monotonically with time.

The important feature in the learning process of nSOM is that only one neuron of all the neurons in the all SOMs can be a winner for one input data, therefore, n SOMs compete with each other. In addition, by the two differences between $\mathrm{SOM_G}$ and $\mathrm{SOM_L}$, $\mathrm{SOM_L}$ tends to self-organize the area where the input data are dense, and $\mathrm{SOM_G}$ tends to self-organize the whole input space except the area occupied by $\mathrm{SOM_L}$. Namely, $\mathrm{SOM_L}$ can find the clusters by themselves.

We have applied nSOM to clustering. After learning of nSOM, in order to extract only clusters from the input data including a lot of noises, we calculate the distance between the input data $\boldsymbol{x}_j$ and the weight vectors in $\mathrm{SOM_L}l$. If the calculated distance is smaller than a threshold value R, the input data $\boldsymbol{x}_j$ is classified into the cluster corresponding to $\mathrm{SOM_L}l$. However, in order to extract clusters using nSOM, we need to know the exact number of clusters in advance.

III. Peace SOM (PSOM)

In this study, we propose the Peace SOM (PSOM) algorithm which possesses both competing and accommodating abilities. We use this algorithm after executing the nSOM method.

A. Learning Algorithm

We explain the learning algorithm of PSOM. In the PSOM algorithm, we use only $\mathrm{SOM_L}$ after executing the nSOM.

(PSOM1) An input data $\boldsymbol{x}_j$ is inputted to all the neurons of $\mathrm{SOM_L}l$ at the same time in parallel.

(PSOM2) We find winner neurons c_l in every $\mathrm{SOM_L}l$ by calculating the distance between $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_{Lli}$ of the neuron i of $\mathrm{SOM_L}l$, according to;

$$c_l = \arg\min_i \{\|\boldsymbol{w}_{Lli} - \boldsymbol{x}_j\|\}. \tag{1}$$

In other words, all SOMs of PSOM have one winner neuron. In this study, Euclidean distance is used for (1).

(PSOM3) The weight vectors of all the neurons of all $\mathrm{SOM_L}l$ are updated according to the following equation;

$$\boldsymbol{w}_{Lli}(t+1) = \boldsymbol{w}_{Lli}(t) + h_{Llc_l,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_{Lli}(t)). \tag{2}$$

The function $h_{Llc_l,i}(t)$ is called the neighborhood function and is one of the best used function in the learning algorithms of SOM. This is described as follows;

$$h_{Llc_l,i}(t) = p_{c_l}(t) \exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_{c_l}\|^2}{2\sigma^2(t)}\right), \tag{3}$$

where $p_{c_l}(t)$ is the learning function, $\boldsymbol{r}_i$ and $\boldsymbol{r}_{c_l}$ are the vectorial locations on the display grid, and $\sigma(t)$ corresponds to the widths of the neighborhood function. $\sigma(t)$ is generally given as decreasing with time according, in this study, it is according to the following equation;

$$\sigma(t) = \sigma(0)(1 - t/T), \tag{4}$$

where T is the maximum number of the learning. The learning function $p_{c_l}(t)$ is a key function deciding the behavior of PSOM and is explained in the next subsection.

(PSOM4) The steps from (PSOM1) to (PSOM3) are repeated for all the input data, namely from $j = 1$ to $j = N$.

B. Learning function

We propose the learning function for PSOM. The learning function plays an important role in the algorithm and produces the competing behavior and the accommodating behavior of PSOM. The value of the learning function is determined by the distance between the input vector $\boldsymbol{x}_j$ and the winner neuron c_l of $\mathrm{SOM_L}l$, according to;

$$p_{c_l}(t) = \alpha(t) \exp\left(-\frac{\|\boldsymbol{w}_{Llc_l} - \boldsymbol{x}_j\|^2}{2\sigma_P^2}\right), \tag{5}$$

where $\boldsymbol{w}_{Llc_l}$ is the weight vector of the winner neuron c_l of $\mathrm{SOM_L}l$, σ_P corresponds to the width of the learning function, and $\alpha(t)$ corresponds to the maximum value of the learning function. $\alpha(t)$ decrease with time according to the following equation;

$$\alpha(t) = \alpha(0)(1 - t/T). \tag{6}$$

It is desirable to choose σ_P as a small value such as less than 0.1, because this is a key point to decide their competing behavior and accommodating behavior. Namely, some SOMs whose winner neuron is near the input data are significantly updated and this means that these SOMs tend to accommodate the input data each other. Conversely, some SOMs whose winner neuron is far apart from the input data are updated very little and this means that these SOMs keep to compete with other SOMs.

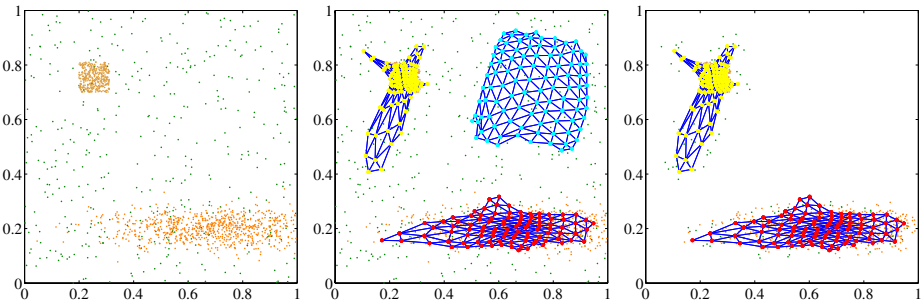
IV. Application to Clustering

A. 2-dimensional input data

First, we consider 2-dimensional input data as shown in Fig. 1(a). The input data is generated artificially as follows. Total number of the input data N is 1600. 25% of the input data are distributed within a range from 0.2 to 0.3 horizontally and from 0.7 to 0.8 vertically, and these data are called the cluster C_1. 50% of the input data are distributed in another cluster C_2, whose horizontal-values follow the normal distribution $N(0.7,\ 0.04)$, and the vertical-values $N(0.2,\ 0.0016)$. The remaining 25% of the input data are distributed between 0 and 1 at random.

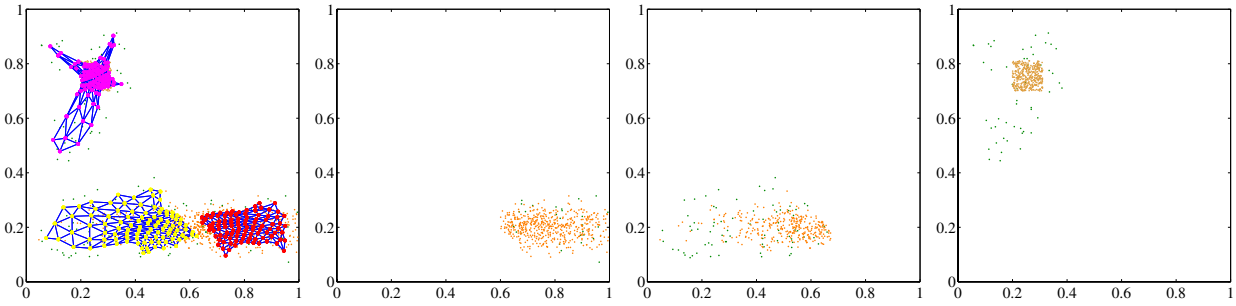
Because the input data include 2 clusters, we use one $\mathrm{SOM_G}$ and two $\mathrm{SOM_L}$ ($n = 3$). Each SOM has 100 neurons (10×10), namely 3 SOMs have totally 300 neurons, and these neurons are arranged on a 2-dimensional hexagon. We repeat the learning four times for all input data.

The simulation result is shown in Fig. 1(b). We can see that two $\mathrm{SOM_L}$ stay around the two clusters by the competing



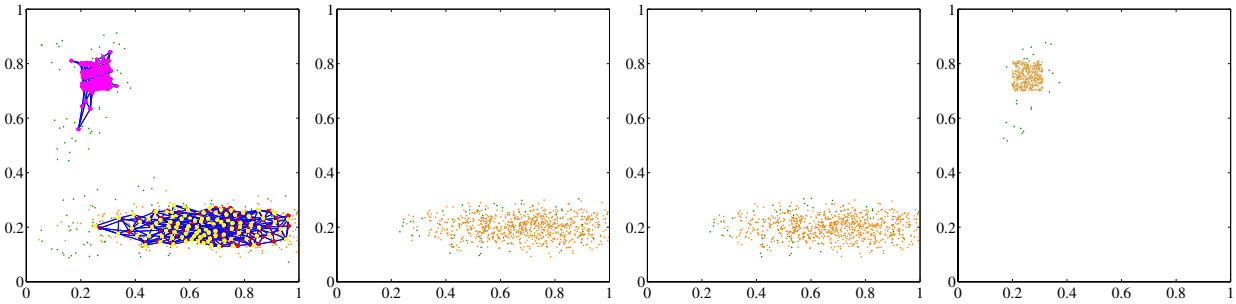
(a) (b) (c)

Fig. 1. Clustering of 2-dimensional input data. (a) Input data. (b) Simulated result of nSOM ($n = 3$). (c) Clusters extracted by $\mathrm{SOM_L}$.



(a) (b) (c) (d)

Fig. 2. Extraction of clusters using inappropriate number of $\mathrm{SOM_L}$. (a) 3 $\mathrm{SOM_L}$ for 2 clusters. (b) Cluster extracted by $\mathrm{SOM_{L1}}$. (c) Cluster extracted by $\mathrm{SOM_{L2}}$. (d) Cluster extracted by $\mathrm{SOM_{L3}}$.



(a) (b) (c) (d)

Fig. 3. Input data extracted by PSOM. (a) Learning result of PSOM ($n = 4$). (b) $\mathrm{SOM_{L1}}$. (c) $\mathrm{SOM_{L2}}$. (d) $\mathrm{SOM_{L3}}$.

behavior of nSOM method. Figure 1(c) shows the two $\mathrm{SOM_L}$ and the input data classified into the clusters corresponding to them using algorithm in the Section II ($R = 0.05$). As we can see from the figures, $\mathrm{SOM_L}$ can successfully extract the clusters, and the noises are removed by $\mathrm{SOM_G}$. This is the result of the nSOM when we know the number of the clusters in advance.

On the other hand, Fig. 2(a) shows the result for the same input data using three $\mathrm{SOM_L}$, namely the number of $\mathrm{SOM_L}$ is more than the number of clusters. In this case two $\mathrm{SOM_L}$ compete with each other in the same cluster located at the lower part of the input space and the extraction of the clusters does not succeed as shown in Figs. 2(b) and (c).

Now, we apply the PSOM algorithm after Fig. 2(a). We repeat the PSOM algorithm four times for all the input data. The parameters of the learning are chosen as follows;

$$\alpha(0) = 0.4, \sigma(0) = 1.5, \sigma_P = 1/16.$$

The result is shown in Fig. 3(a). We can see that the two $\mathrm{SOM_L}$, which competed with each other in Fig. 2(a), are accommodating in one cluster. This is because all SOMs have a winner neuron for one input data and the SOMs whose winner neurons are close to the input data are significantly updated. On the other hand, the $\mathrm{SOM_L}$ situated on the other cluster located close to the left-top corner keeps the position without accommodating. This is because the SOMs whose winner neurons are far apart from the input data are updated very little.

We carry out the extraction of clusters from Fig. 3(a). Figures 3(b), (c) and (d) show that the input data are classified into the clusters corresponding to $\mathrm{SOM_{L1}}$, $\mathrm{SOM_{L2}}$ and $\mathrm{SOM_{L3}}$, respectively ($R = 0.05$).

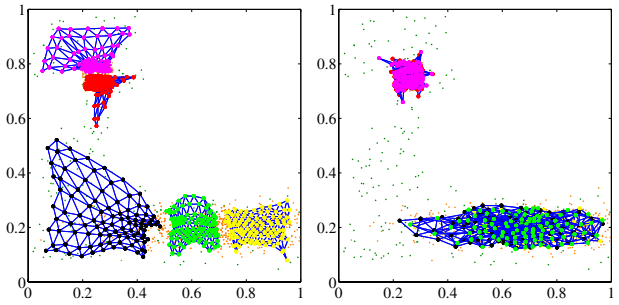
In order to investigate the ability of the PSOM algorithm,

TABLE II

5-DIMENSIONAL GAUSSIAN DATA.

$No.$		Coordinate axes					Probability [%]
		x_1	x_2	x_3	x_4	x_5	
C_1	Mean value	0.8	0.3	0.7	0.2	0.7	15
	Variance	0.0064	0.0324	0.0144	0.0036	0.0064	
C_2	Mean value	0.35	0.8	0.3	0.7	0.2	20
	Variance	0.0144	0.0016	0.04	0.0324	0.0144	
C_3	Mean value	0.6	0.9	0.1	0.1	0.9	15
	Variance	0.0016	0.0064	0.0036	0.01	0.0036	
C_4	Mean value	0.2	0.1	0.8	0.4	0.3	20
	Variance	0.0064	0.04	0.0016	0.09	0.01	

we carry out the simulations with excessive number of SOMs. Figure 4 shows the result using six SOMs for the same input data, namely, 5 $\mathrm{SOM_L}$ just for 2 clusters. As we can see from the figures, 5 $\mathrm{SOM_L}$ can not extract the clusters correctly. However, PSOM can extract 2 clusters as if they know the number of the clusters.



(a) (b)

Fig. 4. Cluster extraction using excessive number of SOMs. (a) Learning result of nSOM ($n = 6$). (b) Learning result of PSOM.

In order to evaluate the clustering ability of PSOM quantitatively, we define the correct answer rate R_{CI} as follows;

$$R_{CI} = \frac{N_{rI} - N_{eI}}{N_{CI}}, \qquad (I = 1, 2, \cdots), \tag{7}$$

where N_{CI} is the true number of the input data within the cluster C_I, N_{rI} is the obtained number of the desired input data within C_I, and N_{eI} is the obtained number of undesired input data out of C_I.

The calculated results are summarized in Table I. When the number of $\mathrm{SOM_L}$ is equal to the number of the clusters (n=3), R_{CI} is similar without adding PSOM algorithm. However, for the case of larger n, the effect of the PSOM is obvious.

TABLE I

CORRECT ANSWER RATE [%] FOR 2-DIMENSIONAL INPUT DATA.

n	Method	$\mathrm{SOM_L}_l$				
		1	2	3	4	5
3	nSOM	86.78	85.07	-	-	-
	PSOM	87.66	90.55	-	-	-
4	nSOM	60.71	32.49	87.31	-	-
	PSOM	88.79	88.92	93.03	-	-
6	nSOM	89.55	47.98	66.92	7.81	41.94
	PSOM	95.27	88.41	94.53	88.29	88.54

B. 5-dimensional input data

Furthermore, we perform the simulation for 5-dimensional input data of 1600 points. This data include four clusters and a lot noises, and the four clusters are generated by random Gaussian data as shown in Table II.

We carry out simulations for the cases of $n = 5$ and $n = 7$. The parameters of the learning for $n = 5$ and $n = 7$ are chosen as follows;

$$n = 5 \Rightarrow \alpha(0) = 0.7,\ \sigma(0) = 1.3, \sigma_P = 1/16,$$
$$n = 7 \Rightarrow \alpha(0) = 0.85, \sigma(0) = 1.7, \sigma_P = 1/16.$$

The results of the calculated correct answer rates are summarized in Table III. We can say that the PSOM algorithm is effective in the case that we do not know the number of the clusters in advance, and is useful to clustering for higher-dimensional input data.

TABLE III

CORRECT ANSWER RATE [%] FOR 5-DIMENSIONAL INPUT DATA.

n	Method	$\mathrm{SOM_L}_l$					
		1	2	3	4	5	6
5	nSOM	95.59	82.65	82.99	90.07	-	-
	PSOM	98.24	81.07	81.33	90.40	-	-
7	nSOM	87.14	80.46	46.48	31.23	71.61	29.14
	PSOM	84.23	87.09	98.24	61.83	75.08	65.23

V. CONCLUSIONS

In this study, we have proposed the Peace SOM (PSOM) algorithm which possesses both competing and accommodating abilities. We have investigated its competing and accommodating behaviors. Furthermore, we have applied PSOM to extract clusters in the case that we do not know the number of the clusters in advance, and have confirmed the efficiency.

REFERENCES

[1] T. Kohonen, *Self-organizing Maps*, Berlin, Springer, vol. 30, 1995.

[2] Y. Cheng, "Clustering with Competing Self-Organizing Maps," *Proc. of IJCNN'92*, vol. IV, pp.785-790, 1992.

[3] W. Wan and D. Fraser, "M2dSOMAP:Clustering and Classification of Remotely Sensed Imagery by Combining Multiple Kohonen Self-Organizing Maps and Associative Memory," *Proc. IJCNN*, vol.3, pp.2464-2467, 1993.

[4] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp.586-600, 2000.

[5] I. Lapidot, H. Guterman and A. Cohen, "Unsupervised Speaker Recognition Based on Competition Between Self-Organizing Maps," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp.877-887, 2002.

[6] H. Matsushita and Y. Nishio, "Competing Behavior of Two Kinds of SOMs and its Application to Clustering," *Proc. WSOM*, pp.355-362, 2005.