Behavior of Community Self-Organizing Map for Clustering and Data Extraction

Taku Haraguchi, Haruna Matsushita and Yoshifumi Nishio
Department of Electrical and Electronic Engineering,
Tokushima University
Email: {taku, haruna, nishio}@ee.tokushima-u.ac.jp

***Abstract*— In the previous study, we have proposed the Community Self-Organizing Map (CSOM) that the neurons create some neuron-community according to their winning frequencies. In this study, we apply CSOM to clustering and data extraction for various input data including a lot of noises, and we investigate its numerical efficiency by using correct answer rate. We confirm that CSOM creates some communities and obtain effective results for data extraction.**

I. Introduction

In data mining, clustering is one of typical analysis techniques and is studied for many applications, such as a statement, a pattern recognition, an image analysis and so on. Then, the Self-Organizing Map (SOM) [1] has attracted attention for the study on clustering in recent years. SOM is an unsupervised neural network introduced by Kohonen in 1982 and is a simplified model of the self-organization process of the brain. SOM obtains statistical feature of input data and is applied to a wide field of data classifications [2]$-$[8]. SOM can classify input data according to similarities and patterns which are obtained by the distance between neurons and some visualization methods based on SOM were proposed [9]-[13]. On the other hand, in the real world, the amount and the complexity of data increase from year to year. Therefore, it is important to investigate various extraction method of clusters from data including a lot of noises.

Meanwhile, the real world is competitive society and human-beings belong to sub-society called “Community”. It is based on definition that the human-beings are social animals introduced by Aristotle. The social animal creates a society and lives in the society. Furthermore, the social animals in the society are centered around a leader and create the community. Additionally, there are also human-beings excluded from the community. The phenomenon actually take place in the real world. On the other hand, the neuron’s world is also competitive society like the real world. Therefore, we consider that the neurons are centered around a leader and create the community. Namely, the neurons are also the social animal. In previous study, we have proposed the Community Self-Organizing Map (CSOM) [14] that the neurons create some neuron-community according to their winning frequency, and the neurons, which is not satisfied the condition, are removed from the community including these after all leaning.

In this study, we apply CSOM to clustering and data extraction for various input data including a lot of noises. Furthermore, we investigate its numerical efficiency by using correct answer rate.

II. Community Self-Organizing Map

In the previous study, we have proposed a Community SOM (CSOM) algorithm. The important feature of CSOM is that the neurons create some neuron-community according to their winning frequency. In other words, in CSOM algorithm, the winner, which satisfies the condition for the winning frequency, and its neighborhood neurons, which satisfy the same condition, create $k_{\rm th}$ community C_k. In the community C_k, a leader l_k is a neuron that has become the winner most frequently among the all neurons belonging to C_k. Because in the human society, the human-beings also creates some community. This phenomenon has tendency that human-beings gather around a leader.

A. Learning Algorithm

We explain the learning algorithm of CSOM in detail. CSOM has a two-layer structure of the input layer and the competitive layer as the conventional SOM. In the input layer, there are d-dimensional input vectors $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$. In the competitive layer, M neurons are arranged as a regular 2-dimensional grid. Each neuron has a weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, M)$ with the same dimension as the input vector. A winning frequency W_i is associated with each neuron and is set to zero initially: $W_i = 0$. The number of members in each community C_k and the number of community n are zero. Before learning, the all neurons do not belong to any community, however, they gradually belong to some community with learning.

(CSOM1) Input an input vector $\boldsymbol{x}_j$ to all the neurons simultaneously in parallel.
(CSOM2) Find a winner c by calculating a distance between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_i$ of each neuron i;

$$c = \arg\min_i \{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}, \tag{1}$$

where $\|\cdot\|$ is the distance measure, in this study, we use Euclidean distance.
(CSOM3) Updated the weight vectors of all the neurons as

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + h_{c,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \tag{2}$$

where t is the learning step. $h_{c,i}(t)$ is called the neighborhood function and is described as

$$h_{c,i}(t) = \alpha(t) \exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2(t)}\right), \quad (3)$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_c$ are the vectorial locations on the display grid, $\alpha(t)$ is called the learning rate, and $\sigma(t)$ corresponds to the width of the neighborhood function. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time, in this study, we use

$$\alpha(t) = \alpha(0)\left(1 - \frac{t}{T}\right), \quad \sigma(t) = \sigma(0)\left(1 - \frac{t}{T}\right), \quad (4)$$

where T is the maximum number of the learning.
If $t \geq T_{\min}$ is satisfied, perform (CSOM4). If not, perform (CSOM9). $T_{\min}$ is fixed parameter and the minimum number of the learning in creating community.
(CSOM4) Increase the winning frequency of the winner c by

$$W_c^{\text{new}} = W_c^{\text{old}} + 1. \quad (5)$$

Evaluate whether the winner c satisfies the conditions of the winning frequency to update the community informations. If $W_c > W_{\text{th}}(t)$ is satisfied, perform (CSOM5). If not, perform (CSOM9) without updating the community. $W_{\text{th}}(t)$ is the threshold value and increases with learning as

$$W_{\text{th}}(t) = (1 - \frac{T_{\min}}{T})\frac{t}{M}. \quad (6)$$

(CSOM5) Find the community C_k including the winner c. If winner c does not belong to any community, create a new community, $n^{\text{new}} = n^{\text{old}} + 1$, and affiliate the winner c to new community C_k as $c \in C_k$ (where $k = n^{\text{new}}$). If not, c remains in its community C_k.
(CSOM6) Find a leader l_k which has become the winner most frequently among the all neurons belonging to C_k, according to Eq. (7) as Fig. 1.

$$l_k = \arg\max_i \{W_i\}, \quad i \in C_k. \quad (7)$$

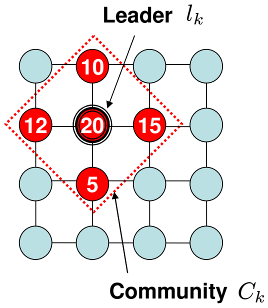

Fig. 1. How to update leader l_k in community C_k. Number in each neuron denotes its winning frequency W_i. The neuron with $W_i = 20$, which is the highest winning frequency among the neurons in the community C_k, becomes the leader l_k.

(CSOM7) Find neurons, whose winning frequency are higher than $W_{\text{th}}(t)$, in 1-neighborhoods of the winner c, then consider whether they belong to any community. If this neighborhood neuron belongs to any community, perform (CSOM8). If not, affiliate it to the community C_k including the winner c in Fig. 2, update the leader l_k as (CSOM6), and perform (CSOM9).

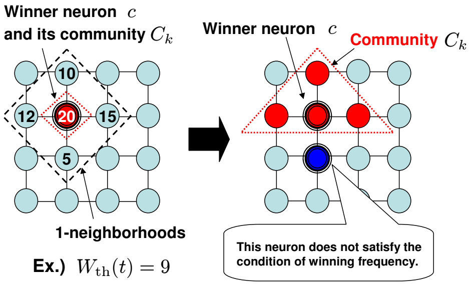

Fig. 2. How to update community C_k. Number in each neuron denotes its winning frequency W_i. The winner's 1-neighborhood neurons with higher winning frequency than W_{th}(t) belong to community C_k. The neuron with $W_i = 5$, which is lower winning frequency than $W_{\text{th}}(t)$, belongs to no community.

(CSOM8) Compare the winning frequencies of two leaders between the community including the winner and the community including winner's neighborhood neuron. Loss of generality, assume that the winner c belongs to C_1 and its neighborhood neuron belongs to C_2. The leaders of C_1 and C_2 are assumed as l_1 and l_2, respectively. If $W_{l_2} \geq W_{l_1}$, the neighborhood neuron keeps on belonging to C_2 as Fig. 3(a). If not, the neighborhood neuron belonging to C_2 are absorbed into C_1 as Fig. 3(b). Then, in a specific case, if the neighborhood neuron is the leader l_2 in the community C_2, all the neurons belonging to C_2 are absorbed into C_1 and decrease the number of communities as $n^{\text{new}} = n^{\text{old}} - 1$.
(CSOM9) Repeat the steps from (CSOM1) to (CSOM8) for all the input data.
(CSOM10) After all learning are finished, check whether $W_i > 0.8 \times W_{\text{th}}(T)$ for each particle i. If it is not satisfied, remove the particle i from the community including it.

III. Experiment Results

A. Application to Clustering

We consider 2-dimensional input data containing three clusters and a lot of noises as shown in Fig. 4(a). The total number of the input data N is 1000, and 200 data are randomly distributed within a range from 0 to 1. The variance of the cluster is different, respectively. The conventional SOM and CSOM have 100 neurons (10×10), respectively. We repeat the learning 15 times for all the input data, namely $T = 15000$. The parameters for the learning of the conventional SOM and CSOM are chosen as follows;

$$\alpha(0) = 0.3, \ \sigma(0) = 3.5, \ T_{\min} = \frac{T}{2}.$$

The simulation results of the conventional SOM and CSOM are shown in Figs. 4(b) and (c), respectively. In Fig. 4(c),

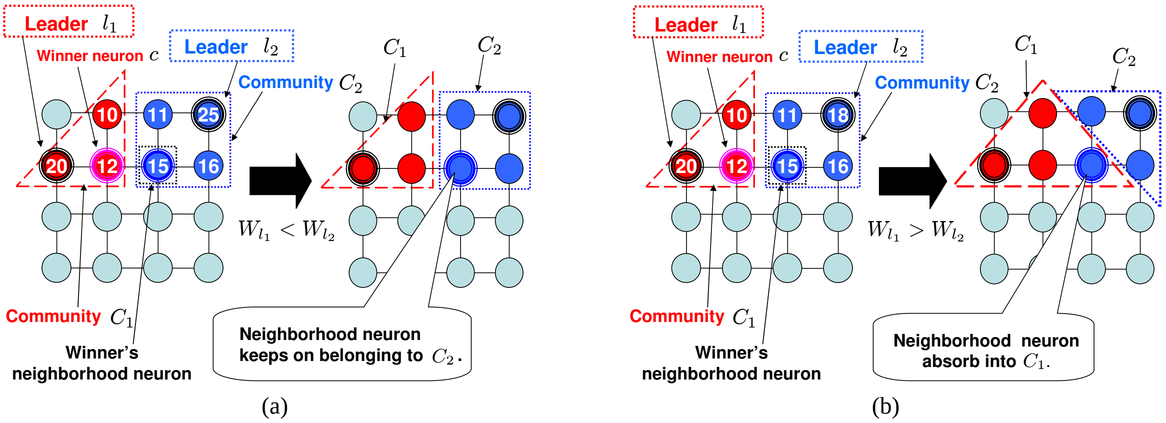

Fig. 3. How to determine whether a community C_1 absorb a community C_2 including a neighborhood neuron. Number in each neuron denotes its winning frequency W_i. The leader l_1 and l_2 are a neuron with the highest winning frequency in the community C_1 and in the community C_2, respectively. (a) As the winning frequency $W_{l_1} = 20$ of the leader l_1 is lower than the winning frequency $W_{l_2} = 25$ of the leader l_2, the neighborhood neuron keeps on belonging to C_2. (b) As $W_{l_1} = 20$ is higher than $W_{l_2} = 18$, the neighborhood neuron is absorbed into C_1.

we can see that the number of communities is three (▲, ★ and ■ mean the each community), and it is the same as the number of clusters. It means that only the neurons, which self-organize the area where the input data are concentrated, create the communities. Furthermore, the neurons belonging to the largest community, which is the largest in the number of neurons belonging to the community, self-organize the largest cluster. Therefore, we can see the number of clusters and the rough condition by investigating the number of communities and the size.

B. Application to Data Extraction

Next, we carry out the extraction of cluster from the results of the conventional SOM and CSOM as Figs. 4(b) and (c). The extraction method is relatively simple as follows. In the conventional SOM, after learning, the input data, which is within a radius of R from all neurons on the map, are classified into the cluster. In CSOM, after learning, the input data, which is within a radius of R from all neurons belonging to each community on the map, are classified into the cluster.

The extraction result of the conventional SOM is shown in Fig. 5(a), and the extraction results by using the respective communities and the only largest community in CSOM are shown in Figs. 5(b) and (c), respectively ($R = 0.05$). In Fig. 5(a), we can see that the cluster obtained by the conventional SOM includes a lot of noises. In other words, the conventional SOM obtains the unnecessary data. In CSOM, as the neurons belonging to any community self-organize any cluster, the results as Fig. 5(b) obtain three clusters and hardly include the noises. Besides, as the neurons self-organizing any cluster create one community at the area, we can obtain the largest cluster by extracting the largest community as Fig. 5(c).

C. Numerical Analysis for Data Extraction

In order to investigate the ability of the conventional SOM and CSOM quantitatively, we define the correct answer rate R_{CI} as follows [8];

$$R_{CI} = \frac{N_{rI} - N_{eI}}{N_{CI}}, \qquad (i = 1, 2, \cdots), \tag{8}$$

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.

TABLE I

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

Method	NeI	NrI	Correct answer rate [%]
Conventional SOM	96	789	86.6
CSOM	43	755	89.0

Table I shows the correct answer rate R_{CI} of the conventional SOM and CSOM for the 2-dimensional data, respectively. From this table, we can see that the correct answer rate R_{CI} and the obtained number of undesired input data N_{eI} of CSOM is better value than them of the conventional SOM. It means that CSOM hardly include the noises and can exactly extract the clusters than the conventional SOM. Therefore, we can that CSOM obtains numerically effective result than the conventional SOM.

IV. Conclusions

In this study, we have applied CSOM to the clustering and the data extraction for various input data including a lot of noises. We have confirmed that the number of communities in CSOM is the same as the number of clusters, and CSOM can obtain visually and numerically effective results for the data extraction.

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.

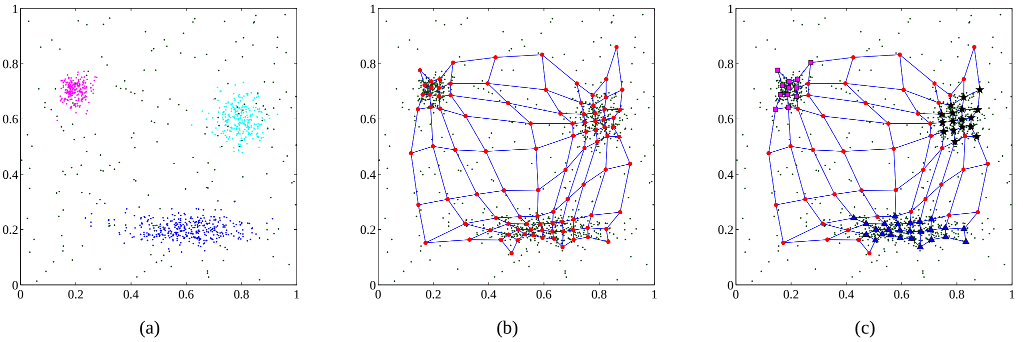

Fig. 4. Learning simulation for 2-dimensional data by using the conventional SOM and CSOM. ▲, ★ and ■ denote the largest community C_1, the second largest community C_2 and the third largest community C_3, respectively. (a) Input data. (b) Learning result of the conventional SOM. (c) Learning result of CSOM.

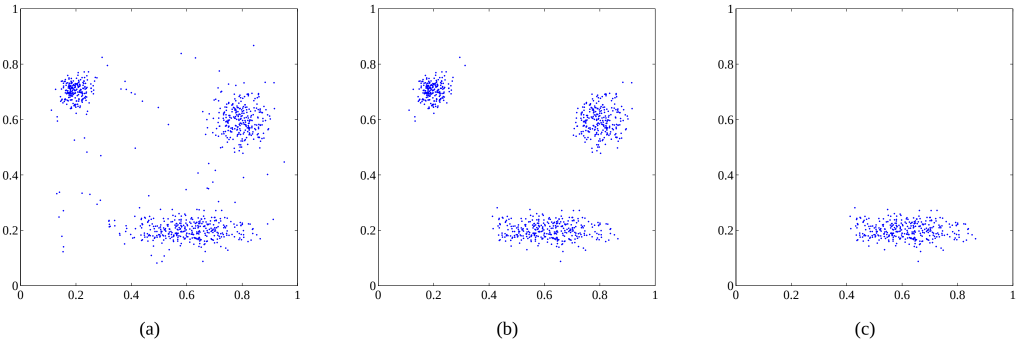

Fig. 5. Extraction results of clusters. (a) Clusters extracted by using all the neurons in the conventional SOM. (b) Clusters extracted by using all the neurons belonging to the respective community in CSOM. (c) Clusters extracted by using the largest community C_1 in CSOM.

[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. of IJCNN’93*, vol. III, pp. 2464-2467, 1993.

[4] J. Vesanto and E. Alhoniemi, “Clustering of the Self-Organizing Map,” *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 586–600, 2002.

[5] C. Derek and E. Adrian, “Finding Curvilinear Features in Spatial Point Patterns: Principal Curve Clustering with Noise,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 601-609, 2000.

[6] P. Doucette, P. Agouris and A. Stefanidis, “Self-Organized Clustering for Road Extraction in Classified Imagery,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 55, Issues 5-6, pp. 347-358, 2001.

[7] A. Ultsch, “Clustering with SOM: U*C,” *Proc. Workshop on Self-Organizing Maps.*, pp.75-82, 2005.

[8] H. Matsushita and Y. Nishio, “Tentacled Self-Organizing Map for Effective Data Extraction,” *Proc. International Neural Network Conference on Neural Networks*, pp. 1929-1936, 2006.

[9] A. Ultsch and H. P. Siemon, “Kohonen’s Self Organizing Feature Maps for Exploratory Data Analysis,” *Proc. International Neural Network Conference*, pp. 305-308, 1990.

[10] X. Zhang and Y. Li, “Self-organizing map as a new method for clustering and data analysis,” *Proc. International Neural Network Conference on Neural Networks*, pp. 2448-2451, 1993.

[11] M. A. Kraaijveld, J. Mao, and A. K. Jain, “A nonlinear projection method based on Kohonen’s topology preserving maps,” *IEEE Trans. Neural Netws.*, vol. 6, no. 3, pp. 548-559, 1995.

[12] N. R. Pal and V. K. Eluri, “Two efficient connectionist schemes for structure preserving dimensionality reduction,” *IEEE Trans. Neural Netws.*, vol. 9, no. 6, pp. 1142-1154, 1998.

[13] M. C. Su and H. T. Chang, “A new model of self-organizing neural networks and its application in data projection,” *IEEE Trans. Neural Netws.*, vol. 12, no. 1, pp. 153-158, 2001.

[14] T. Haraguchi, H. Matsushita and Y. Nishio, “Community Self-Organizing Map and its Application to Data Extraction,” *Proc. International Neural Network Conference on Neural Networks*, June. 2009.