2009 International Workshop on Nonlinear Maps and their Applications
NOMA'09, Urbino, Italy, September 10-11, 2009 (paper id: 008)

Behavior of Community Self-Organizing Map with Camaraderie

Taku Haraguchi
Dept. Electrical and Electronic Eng.,
Tokushima University
Email: taku@ee.tokushima-u.ac.jp

Haruna Matsushita
Dept. Electrical and Electronic Eng.,
Tokushima University
Email: haruna@ee.tokushima-u.ac.jp

Yoshifumi Nishio
Dept. Electrical and Electronic Eng.
Tokushima University
Email: 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 modify the algorithm of CSOM and propose CSOM with camaraderie. As neurons belonging to any community have the camaraderie in the community, they tend to attract each other. Therefore, the neurons belonging to any community self-organize only concentrated part of input data including a lot of noises. We apply modify CSOM and CSOM with camaraderie to clustering problem for clustering and data extraction, and investigate its behavior and its effectiveness. The efficiencies of COM with camaraderie are confirmed by several results.**

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]-[5]. 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. 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, it is based on definition that human-beings are social animals introduced by Aristotle. The human-beings a easily gather around the leader. In other words, the community is created as core on the leader of the community. In previous study, we have proposed the Community Self-Organizing Map (CSOM) [6] that the neurons create some neuron-community according to their winning frequency. We have applied CSOM for clustering and data extraction to various input data, and confirmed the efficiency of CSOM. However, as the simulation result of CSOM is the same as the simulation result of the conventional SOM, it is difficult to exactly extract the clusters from data including a lot of noises.

In this study, we modify the algorithm of CSOM and propose CSOM with camaraderie. The features of CSOM with camaraderie are that the neurons belonging to any community have the camaraderie in the community. Therefore, the neurons belonging to any community self-organize the only concentrated input data for the input data including a lot of noises. We apply the modified CSOM and CSOM with camaraderie to clustering problem for clustering, gray scale display method and data extraction, and investigate its behavior and the its effectiveness.

II. Modified Community Self-Organizing Map

In our previous research, we have proposed CSOM that the neurons create some neuron-community according to their winning frequency, called community learning, and we have confirmed the number of communities is the same as the number of clusters. However, the neurons, which self-organize the the area where the input data are concentrated, can not frequently create the exact communities. Because there is a significant difference in the winning frequency of the the neurons located outside between other neurons. Therefore, in most cases, CSOM can not exactly perform the community learning.

In this study, we modify the algorithm of CSOM as follows. All the neurons start to accumulate the winning frequency when the neurons start to create the community. Because, when the neurons start to accumulate the winning frequency after certain steps of the leaning, there is little difference in the winning frequency of the the neurons located outside between other neurons. As the results, the flexibility of the community learning in the modified CSOM enhances. Furthermore, in order to enhance the flexibility of the modify CSOM we propose CSOM with camaraderie. The features of CSOM with camaraderie are that the neurons belonging to any community have the camaraderie and attract each other in the community.

A. Learning Algorithm

We explain the learning algorithm of the modified 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

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

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

$$h_{Cc,i}(t) = \begin{cases} \beta(t)\exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2(t)}\right), & \text{if } i \in C_k \\ \alpha(t)\exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2(t)}\right), & \text{otherwise,} \end{cases} \tag{3}$$

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. $\sigma(t)$ decrease monotonically with time;

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

where T is the maximum number of the learning. The learning function $\beta(t)$ is explained in the next subfunction.

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

$$W_c^{\text{new}} = W_c^{\text{old}} + 1. \tag{5}$$

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

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

(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.

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

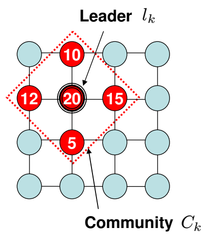


Fig. 1. How to update leader l_k in community C_k. Number in each neuron denotes its winning frequency W_i. The neuron with $W_i = 20$, which is the highest winning frequency among the neurons in the community C_k, becomes the leader l_k.

(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).

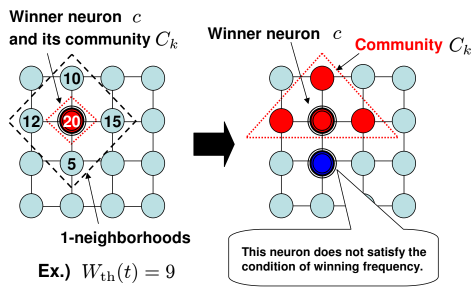


Fig. 2. How to update community C_k. Number in each neuron denotes its winning frequency W_i. The winner's 1-neighborhood neurons with higher winning frequency than W_{th}(t) belong to community C_k. The neuron with $W_i = 5$, which is lower winning frequency than $W_{\text{th}}(t)$, belongs to no community.

(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. If not, the neighborhood neuron belonging to C_2 are absorbed into C_1. 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 $W_i > 3T/4M$ for each particle i. If it is not satisfied, remove the particle i from the community including it.

B. Learning function

We propose new learning function based on the learning function proposed in our past study [5]. The value of the learning function is determined by the distance between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_{Ci}$ of the neuron i belonging to the any community according to fellowing equation;

$$\beta(t) = \alpha(t) \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{r}_i\|^2}{2\sigma_C^2(t)}\right), \tag{8}$$

where σ_C is a fixed parameter and is called a camaraderie parameter. The camaraderie parameter controls the camaraderie of the neurons belonging to the any community. If the parameter σ_C is more small value, the movement of the neurons, which are away from the input data, belonging to the any community weakens. In this case, we can say that the camaraderie in the community enhances.

III. Experiment Results

A. Comparison with three algorithms

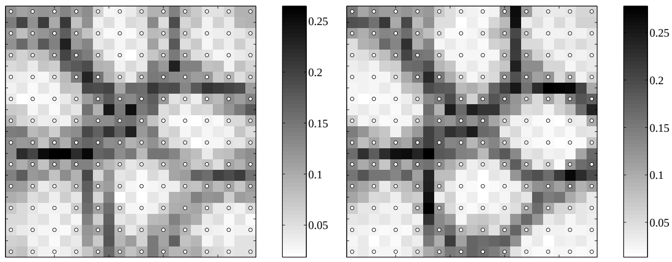
We consider the 2-dimensional input data containing seven clusters and a lot of noises as shown in Fig. 3(a). The total number of the input data N is 1000 and 200 data are randomly distributed within a range from 0 to 1. The respective clusters have 800 data, and the number of each cluster and the variance of all the clusters are about the same values. The conventional SOM, the modified CSOM and CSOM with camaraderie have 144 neurons (12×12), respectively. We repeat the learning 15 times for all the input data, namely $T = 15000$. The parameters for the learning for two algorithm are chosen as follows;

$$\alpha(0) = 0.3,\ \sigma(0) = 3.5,\ T_{\min} = \frac{T}{2},\ \sigma_C = \frac{1}{15}.$$

Figures 3(b), (c) and (d) show the learning results of the conventional SOM, the modified CSOM and CSOM with camaraderie, respectively. In Figs. 3(b) and (c), we can see that the conventional SOM is easily affected to the noises. In addition, the simulation result of the modified CSOM is the same as the simulation result of the conventional SOM. Therefore, the modified CSOM is also affected to the noises and the neurons, which self-organize the unnecessary data, also belong to any community. On the other hand, in Fig. 3(d), we can see that CSOM with camaraderie is not affected to the noises and the number of communities are the same as the number of clusters. This means that the neurons belonging to any community self-organize only concentrated part of the input data. Let us consider this obtained result. In CSOM with camaraderie, the neurons belonging to the any community move to fit the area where the input data are concentrated. The behavior can be explained by the learning function Eq. 8. This learning function enhances the movement of the neurons if the neurons belonging to the any community are close to the input data. Meanwhile, this learning function weaken the movement of the neurons if the neurons belonging to the any community are away from the input data. Hence, the neurons belonging to the any community do not spread out of the area where the input data are concentrated. In other words, the neurons belonging any community have camaraderie and attract each other in the community. From these reasons, we can confirm that CSOM with camaraderie obtains the most effective result.

B. Visualization Method

Next, we confirm the recognition of the input data. As the simulation result of the modified CSOM is the same as the simulation result of the conventional SOM, we use the gray scale display method [8] for the conventional SOM and CSOM with camaraderie as Figs. 4(a) and (b), respectively. From these results, we can see that the boundary lines of CSOM with camaraderie are clearer than the conventional SOM. This means that the total number of neurons self-organizing the respective clusters in CSOM with camaraderie is more than it in the conventional SOM. Therefore, we can say that CSOM with camaraderie obtains more exact map reflecting the distribution state of the input data.



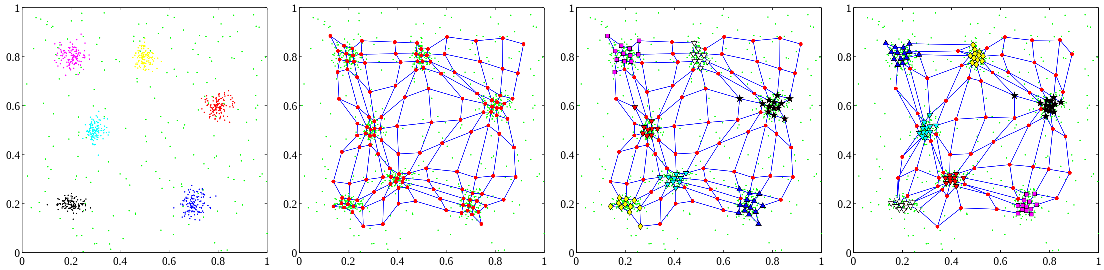
(a) (b)

Fig. 4. Visualization of results of 2-dimensional data. (a) Conventional SOM. (b) CSOM with camaraderie.

C. Application of data extraction

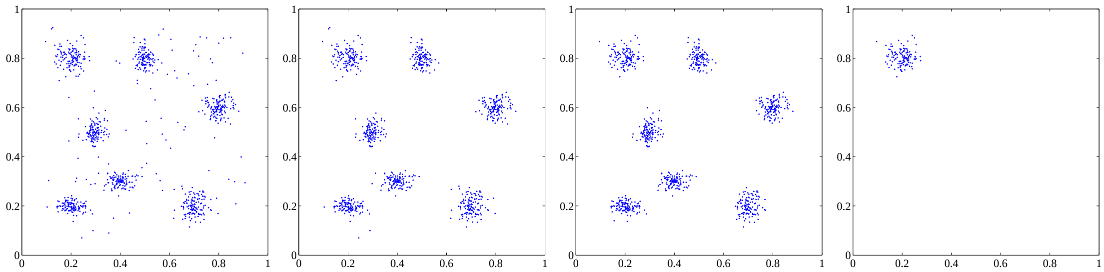
Next, we carry out the extraction of cluster from the results of three algorithm as Figs. 3(b), (c) and (d). The extraction method is relatively simple as follows. In the conventional SOM, after learning, the input data, which is within a radius of R from all neurons on the map, are classified into the cluster. In the modified CSOM and CSOM with camaraderie, 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 of respective communities in the modified CSOM and CSOM with camaraderie are shown in Figs. 5(b) and (c), respectively ($R = 0.05$). In the conventional SOM, 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 the modified CSOM and CSOM with camaraderie, as all the neurons belonging to the each community self-organize the each cluster, the results as Figs. 5(b) and (c) obtain seven clusters and doesn't include a lot of noises. Besides, as all the neurons belonging to the each community self-organize



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

Fig. 3. Simulation results for 2-dimensional data. (a) Input data. (b) Conventional SOM. (c) Conventional CSOM. (d) CSOM with camaraderie.



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

Fig. 5. Extraction results of clusters. (a) Clusters extracted by the conventional SOM. (b) Clusters extracted by the modify CSOM. (c) Clusters extracted by CSOM with camaraderie. (d) Clusters extracted by the largest community C_1 by using CSOM with camaraderie.

the each cluster, we can obtain one cluster by extracting one community, as Fig. 5(d).

Furthermore, in order to investigate the ability of three algorithms quantitatively, we define the correct answer rate R_{CI} as follows [5];

$$R_{CI} = \frac{N_{rI} - N_{eI}}{N_{CI}}, \qquad (i = 1, 2, \cdots), \tag{9}$$

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	115	915	85.6
CSOM	49	847	93.6
CSOM with camaraderie	38	827	94.3

Table I shows the correct answer rate of the conventional SOM, the modified CSOM and CSOM with camaraderie for the 2-dimensional data, respectively. From this table, we can see that the correct answer rate of CSOM with camaraderie is the best value and N_{eI} is smallest in three algorithms. Therefore, we can confirm that CSOM obtains the most exact clusters and the most effective result in three algorithms.

IV. CONCLUSION

In this study, we have modified the algorithm of COM and proposed CSOM with camaraderie. The features of CSOM with camaraderie are that the neurons belonging to any community have the camaraderie in the community. We have applied the modify COM and CSOM with camaraderie to clustering problem, and investigated the efficiency of new algorithm and the camaraderie in the community by using various method. In consequence, we can say that CSOM with camaraderie obtains the most result in three algorithms.

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

[4] 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.

[5] 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.

[6] 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 (accepted).

[7] 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.

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