NOLTA 2006
11-14 september
Bologna - ITALY

Shooting SOM and its Application for Clustering

Masato Tomita†, Haruna Matsushita† and Yoshifumi Nishio†

Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Phone:+81-88-656-7470, FAX:+81-88-656-7471
Email: {tomita, haruna, nishio}@ee.tokushima-u.ac.jp

Abstract—The Self-Organizing Map (SOM) is popular algorithm for unsupervised learning and visualization introduced by Teuvo Kohonen. One of the most attractive applications of SOM is clustering and several algorithms for various kinds of clustering problems have been reported and investigated. In this study, we propose a new type of SOM algorithm, which is called Shooting SOM (SSOM) algorithm. The important feature of SSOM is that the neurons move like aiming at a target, namely, some neurons find near the center of the area where input data are concentrated and other neurons get away from the former neurons. Because of this feature, SSOM tends to self-organize only in the area where input data are concentrated. We investigate the behavior of SSOM and apply SSOM to clustering problems.

1. Introduction

In data mining, clustering is one of the most important issues and is also very useful for many applications, such as industrial engineering, image processing, biology and medicine. Then, the Self-Organizing Map (SOM) attracts attentions for clustering in recent years. SOM is popular tools for clustering and visualization of high-dimensional data [1]. SOM is unsupervised neural network introduced by Kohonen in 1982 [2] and is a model simplifying self-organization process of the brain. SOM maps multidimensional data onto a 2-dimensional grid. SOM can classify input data according to similarities and patterns which are obtained by the distance between neurons and is applied to wide fields of data classifications. Although many methods to extract clusters by using SOM have been proposed [3]-[10], it seems to be very difficult to construct a simple method using SOM for universal input data. On the one hand, in the world, the amount and complexity of data increase from year to year and control of large-volume data get a lot of attention. Therefore, it is important to investigate various extraction methods of some clusters from data including some noises.

In this study, we propose a new type of SOM algorithm, which is called Shooting SOM (SSOM) algorithm. The important feature of SSOM is that the neurons move like aiming at a target, namely, some neurons find near the center of the area where input data are concentrated and other neurons get away from the former neurons. Because of this feature, SSOM tends to self-organize only in the area where input data are concentrated. We investigate the behavior of SSOM and apply SSOM to clustering problems. The efficiencies of SSOM are confirmed by several simulation results.

In the Section II, the algorithm of the conventional SOM is introduced. In the Section III, we explain the learning algorithm of SSOM algorithm. In the Section IV, the behavior of SSOM is explained with some simulation results of clustering.

2. Self-Organizing Map (SOM)

In this section we introduce the conventional SOM algorithm in order to make clear new points of the proposed algorithm.

SOM has two-layer structure of the input layer and the competitive layer. In the competitive layer, m neurons are arranged as a regular 2-dimensional grid. The range of the elements of d-dimensional input data $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$ are assumed.

(SOM1) The initial values of all the weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, m)$ of the neurons are given between 0 and 1 in a reticular pattern.
(SOM2) An input data $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(SOM3) We find the winner neuron c by calculating the distances between $\boldsymbol{x}_j$ and $\boldsymbol{w}_i$ according to;

$$c = \arg\min_i\{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}. \quad (1)$$

In other words, the winner neuron c is the neuron with the weight vector nearest to the input vector $\boldsymbol{x}_j$. In this study, Euclidean distance is used for Eq. (1).
(SOM4) The weight vectors of all the neurons are updated as;

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

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

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

where $\alpha(t)$ is the learning rate, $\boldsymbol{r}_i$ and $\boldsymbol{r}_c$ are the vectorial locations of the neurons on the display grid, and $\sigma(t)$ corresponds to the widths of the neighborhood function. $\alpha(t)$ and $\sigma(t)$ decrease with time according to the following equations;

$$\begin{aligned} \alpha(t) &= \alpha(0)\left(1 - t/T\right), \\ \sigma(t) &= \sigma(0)\left(1 - t/T\right), \end{aligned} \qquad (4)$$

where T is the maximum number of the learning.
(SOM5) The steps from (SOM2) to (SOM4) are repeated for all the input data, namely, from $j = 1$ to $j = N$.

3. Shooting SOM (SSOM)

In this section we introduce SSOM algorithm in order to make clear new points of the proposed algorithm.

3.1. Learning Algorithm

We explain the learning algorithm of SSOM. In SSOM, m neurons are arranged as a regular 2-dimensional grid. The range of the elements of d-dimensional input data $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$ are assumed.

(SSOM1) The initial values of all the weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, m)$ of the neurons are given between 0 and 1 in a reticular pattern.
(SSOM2) An input data $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(SSOM3) We find the winner neuron c by calculating the distances between $\boldsymbol{x}_j$ and $\boldsymbol{w}_i$ according to;

$$c = \arg\min_i\{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}. \qquad (5)$$

In other words, the winner neuron c is the neuron with the weight vector nearest to the input vector $\boldsymbol{x}_j$. In this study, Euclidean distance is used for Eq. (5).
(SSOM4) We measure whether the winner neuron or 1-neighborhood of the winner neuron or otherwise and show in Fig. 1.

Furthermore, we determine the update rate A_i for each cases as follows.;

$$A_i = \begin{cases} 1, & \text{if } i = c \\ -0.04, & \text{if } i \text{ is 1-neighborhood of } c \\ 0, & \text{otherwise.} \end{cases} \qquad (6)$$

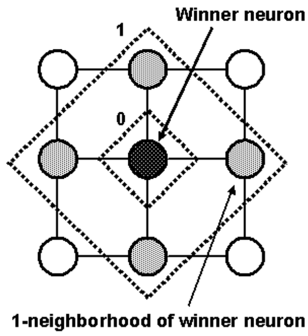

Figure 1: Winner neuron and 1-neighborhood neurons of the winner neuron.

(SSOM5) The weight vectors of all the neurons are updated as;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + A_i \times h_{c,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \qquad (7)$$

where t is the learning step and $\alpha(t)$ is the learning rate. $\alpha(t)$ decrease with time according to the following equations;

$$\alpha(t) = \alpha(0)\left(1 - t/T\right), \qquad (8)$$

where T is the maximum number of the learning.
(SSOM6) The steps from (SSOM2) to (SSOM5) are repeated for all the input data, namely, from $j = 1$ to $j = N$.

4. Simulation Results

4.1. Behavior of SSOM

We consider the 2-dimensional input data of 480 points whose distribution is as Fig. 2. Each 10 clusters distribute 40 points at random.

The remaining 80 points are uniformly distributed

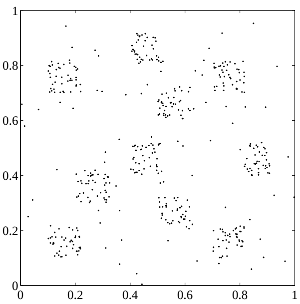

Figure 2: 2-dimensional input data.

between 0 and 1 at random. We consider the conventional SOM and the proposed SSOM with 25 neurons

(5×5). The parameters for the learning of the conventional SOM and SSOM are chosen as follows;

$$\alpha(0) = 0.4,\ \sigma(0) = 1/16, \tag{9}$$

We execute the learning for all input data once. The simulation results of SSOM is shown in Fig. 3(c). SSOM self-organizes near the center of the area where the input data are concentrated.

Let us examine the behavior of SSOM in more detail. As we can see from Fig. 3(b) and Fig. 3(c), the conventional SOM self-organizes input data including some noises with all neurons. However, we can see that SSOM moves like aiming at a target from learning process. Certain neuron moves toward the cluster to hit a center of the area where input data are concentrated, and other neurons moves toward other cluster and go away. Therefore, only one neuron by the one cluster self-organizes its cluster and SSOM can effectively self-organize many clusters. In doing so, SSOM can self-organize as well as the conventional SOM with minimum neurons because SSOM does not use extra neurons.

4.2. Data Extraction

The concept using SSOM can be exploited to extract the data only in clusters of the input data including some noises, because SSOM can find near the center of such areas by itself. We carry out the extraction of cluster after Fig. 3(d). The extraction method is a relatively simple as follows. After learning, the input data which is within a radius of R from all neurons on the map are classified into the cluster and extract only the cluster.

The simulation results of SSOM are shown in Figs. 4(a) and (b), respectively ($R = 0.06$). As we can see from these figures, SSOM can successfully find near the center of the cluster and can extract as well as the conventional SOM with less neurons.

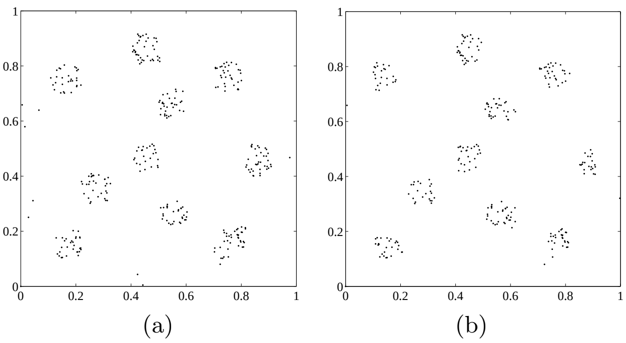

Figure 4: Extraction of cluster. (a) Clusters extracted by the conventional SOM. (b) Clusters extracted by SSOM

4.3. Simulation Results for Large Input Data

We consider the 2-dimensional input data of 1240 points whose distribution is as Fig. 5. Each 31 clusters distribute 40 points at random.

We consider SSOM with 64 neurons (8×8). The parameters for the learning of SSOM are chosen as follows;

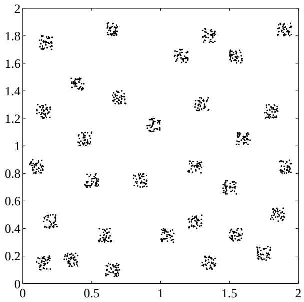

Figure 5: 2-dimensional input data.

$$\alpha(0) = 0.3,\ \sigma(0) = 1/16. \tag{10}$$

We execute the learning for all input data once. The simulation results of SSOM is shown in Fig. 6(a). SSOM self-organizes near the center of the area where the input data are concentrated. Furthermore, we carry out the extraction of cluster after Fig. 6(a). The extraction results of SSOM are shown in Figs. 6(b), ($R = 0.06$). As we can see from these figures, SSOM can successfully find the cluster.

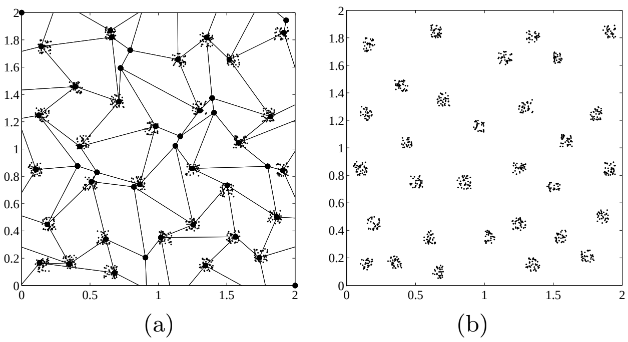

Figure 6: Learning results. (a) Learning result of SSOM. (b) Clusters extracted by SSOM.

5. Conclusions

In this study, we have proposed the Shooting SOM (SSOM). We have explained the differences between the conventional SOM and SSOM have investigated its behavior. Furthermore, we have applied SSOM to extract all near the center of clusters and have confirmed its efficiency.

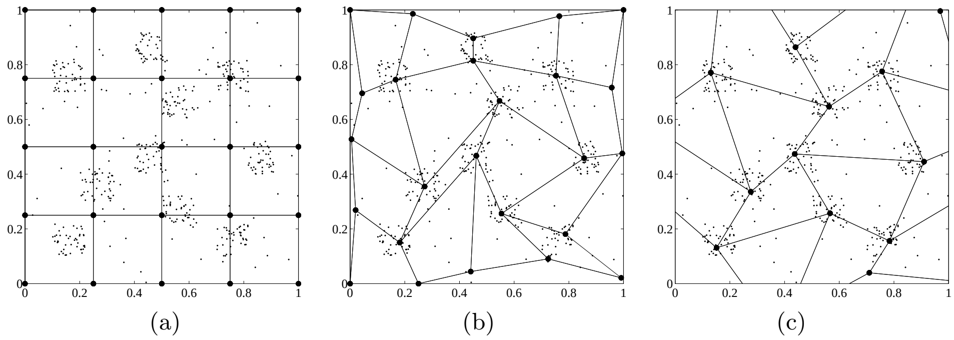

Figure 3: Learning results. (a) Initial state. (b) Learning result of the conventional SOM. (c) Learning result of SSOM.

In the future, we try to discover new applications of SSOM in diverse fields such as sound data processing.

References

[1] T. Kohonen, "The Self-Organizing Maps," *Neurocomputing*, vol. 21, pp. 1-6, 1998.

[2] T. Kohonen, *Self-Organizing Maps*, Berlin, Springer, vol. 30, 1995.

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

[4] 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.

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

[6] 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.

[7] 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.

[8] M. Gervey and S. Lek, "Patterning and Clustering Ecological Assemblages," *Proc. of WSOM'05*, pp. 9-16, 2005.

[9] H. Matsushita and Y. Nishio, "Competing Behavior of Two Kinds of SOMs and its Application to Clustering," *Proc. of WSOM'05*, pp. 355-362, 2005.

[10] H. Matsushita and Y. Nishio, "Competing and Accommodating Behaviors of Peace SOM," *Proc. of ISCAS'06*, 2006 (to appear).