17-3

Clustering of Lazy Self-Organizing Map Considering Lazy-Neuron Rate

Taku HARAGUCHI (Tokushima University) **Haruna MATSUSHITA** (Tokushima University) **Yoshifumi NISHIO** (Tokushima University)

1. Introduction

The Self-Organizing Map (SOM)[1] has attracted attention for the study on clustering in recent years. Meanwhile, there is a fascinating report that about 20% of worker ants are "lazy"[2], however, there is no evidence to prove definite reason and assuredness of the report at the present stage. In this study, we propose Lazy Self-Organizing Map which is a new SOM algorithm. The important feature of the LSOM is three kinds of neurons exist. Furthermore, the learning rate of the LSOM depends on each neuron's character and lazy-neuron rate, and monotonically decreases with the learning step. We apply the LSOM considering lazy-neuron rate to various input data set and confirm its effectiveness.

2. Lazy Self-Organizing Map

In the LSOM algorithm, the feature of the LSOM is that three kinds of neurons exist; *worker neurons*, *lazy neurons*, which do not work, and *indecisive neurons* which are the neighborhoods of the lazy neurons. M neurons have d-dimensional weight vector $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, M)$ respectively. $(p \times M)$ neurons are classified into a set of the lazy neurons S_{lazy} at random. i.e., p denotes lazy-neuron rate.

(LSOM1) An input $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$ is given to all the neurons at the same time in parallel.
(LSOM2) We find a winner neuron c by calculating the distances between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_i$ of the neuron i. If $c \in S_{\mathrm{lazy}}$, we perform (LSOM3). If not, we perform (LSOM4).
(LSOM3) The lazy neuron, which is the winner c, is excluded from the set of the lazy neuron S_{lazy}, and a neuron f is selected to become a member of S_{lazy}. f is farthest from the input data $\boldsymbol{x}_j$ and is not in S_{lazy};

$$f = \arg\max_i \{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}, \quad i \notin S_{\mathrm{lazy}}. \tag{1}$$

In other words, the lazy neuron becomes worker if it becomes the winner c, and another neuron f becomes lazy.
(LSOM4) We find the indecisive neurons. A set of the indecisive neurons N_{lazy} are the neighborhoods of each lazy neuron l in S_{lazy}.

$$\begin{aligned} N_{\mathrm{lazy}} = \{i \mid \quad & \|\boldsymbol{r}_i - \boldsymbol{r}_l\|^2 \le D(t), \\ & i \neq c,\ i \notin S_{\mathrm{lazy}},\ l \in S_{\mathrm{lazy}}\}, \end{aligned} \tag{2}$$

where $||\boldsymbol{r}_i - \boldsymbol{r}_l||$ is the neighborhood distance between map nodes i and l on the map grid, and $D(t)$ corresponds to the neighborhood size. $D(t)$ increases with learning step according to the following equation;

$$D(t) = \left[D_{\max} \frac{t}{T} \right], \tag{3}$$

where $[\ \cdot\ ]$ denotes the Gauss' notation and $D_{\max}$ is a fixed parameter deciding the max value of $D(t)$, t is the learning step, T is the maximum number of the learning.
(LSOM5) The weight vectors of all the neurons are updated. The learning rate of the LSOM $\alpha(t)$ depends on each neuron's character and the lazy-neuron rate p, and monotonically decreases with the learning step;

$$\alpha(t) = \begin{cases} \alpha_{\mathrm{lazy}} \left(1 - p\frac{t}{T}\right), & \text{if } i = l,\ l \in S_{\mathrm{lazy}} \\ \alpha_{\mathrm{N}} \left(1 - (1-p)\frac{t}{T}\right), & \text{if } i \in N_{\mathrm{lazy}} \\ \alpha_{\mathrm{w}} \left(1 - (1-p)\frac{t}{T}\right), & \text{otherwise,} \end{cases} \tag{4}$$

where α_{w} is the learning rate of the worker neurons, α_{lazy} is the learning rate of the lazy neurons and α_{N} is the learning rate of the indecisive neurons, namely $\alpha_{\mathrm{lazy}} < \alpha_{\mathrm{N}} < \alpha_{\mathrm{w}}$.
(LSOM6) The steps from (LSOM1) to (LSOM5) are repeated for all the input data.

3. Application to Clustering

We consider 2-dimensional input data: Target data set shown in Fig. 1(a). The simulation result of the conventional SOM is shown in Fig. 1(b). We can see that the conventional SOM does not self-organize up to all the outliers of the input data. Meanwhile, the result of the LSOM are shown in Fig. 1(c). We can see that the LSOM can self-organize up to all the corner data than the conventional SOM. From these figures, we can confirm the effectiveness of the LSOM for input data having density difference.

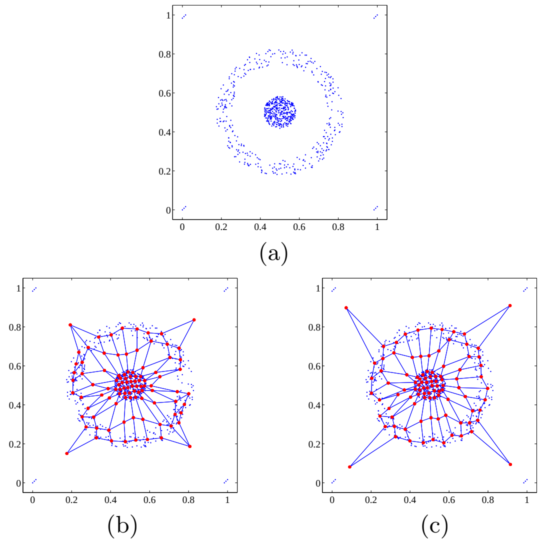

Figure 1: Simulation results for Target data. (a) Input Data. (b) Conventional SOM. (c) LSOM.

4. Conclusions

In this study, we have proposed the LSOM. We have investigated its behaviors and have confirmed the effectiveness of the LSOM containing the lazy neurons, which is from 10% to 20% of the total.

Reference

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

[2] H. Hasegawa, "Optimization of GROUP Behavior," Japan Ethological Society Newsletter, no. 43, pp. 22–23, 2004.