2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Application of growing self-organizing map to small-world networking

Takeshi Ehara[†] **and Toshimichi Saito**[‡]

†, ‡EECE Dept, HOSEI University,Koganei-shi, Tokyo, 184-8584, Japan
Email: ehara@nonlinear.k.hosei.ac.jp, tsaito@k.hosei.ac.jp

Abstract—This paper studies a novel application of growing self-organizing maps to networking. In our algorithm nodes for the networking are applied successively as input data. Adapting to the input, the map can grow and can change the topology. Performing basic numerical experiments, we have confirmed that our algorithm can generate small-world like networks characterized by relatively small average path length and large clustering coefficient. These results provide basic information to develop interesting applications of unsupervised learning algorithms.

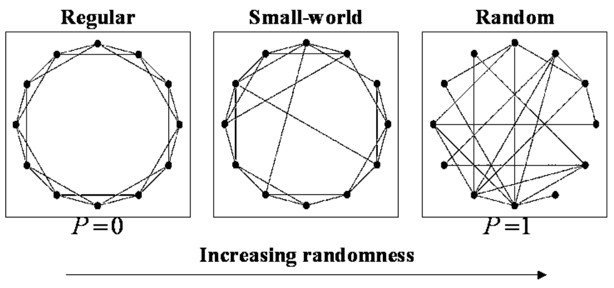

Figure 1: Typical networks.

1. Introduction

The self-organizing feature map (ab. SOM [1]) is known as a typical unsupervised learning system that can extract features of input data. In order to improve adaptability to environment, SOM with growing cell structure (ab. GCS [2], [3]) has been studied. The GCS can change size and topology of maps depending on features of input data. Applications of GCSs are many, including data visualization, pattern classification, vector quantization, traveling sales person problem and image skeletonization [2]-[8].

This paper studies a novel application of the GCS to networking represented by small-world networks (ab. SWNs [9]-[12]). Fig. 1 illustrates feature of the SWN.

The regular network is characterized by connection of each node with its four neighbors. In the algorithm in [9] (WS algorithm) edges of the regular network are reconnected by the probability P without changing the total number of edges. If $P = 0$, the regular network does not change. If $P = 1$, we obtain the random network. Between them we have obtained a SWN that has middle feature of the regular and random networks. In [9] the SWN structure is characterized by two basic measures: small average path length L as regular graph and large clustering coefficient C as random graph [9]. The SWNs can function even if some nodes are out-of-use and can propagate information via short paths. Such networks are probably generic for many natural networks including power grid of a nation, the collaboration graph of film actors and communication with fewer connections [9]-[12]. Also, coupled dynamical system having small-world like topology displays enhanced signal-propagation speed and synchronizability [10]. In order to study such a network structure, some algorithm is necessary to generate a desired network for a set of nodes. In the WS algorithm, given regular lattices are reconnected following some probability without learning.

As compared with the WS algorithm our algorithm has adaptability and flexibility. Nodes for networking are applied successively as input data. The algorithm has subroutines to increase and decrease the number of nodes depending on the learning history with some control parameters. That is, our algorithm can adapt to dynamic input space.

Performing two basic numerical experiments we show that the algorithm can generate SWNs if parameters are selected suitably. These results provide basic information to develop interesting applications of unsupervised learning algorithms including communications system and traffic network and so on.

2. Learning Algorithm

In order to describe the algorithm, we give some basic definitions. Let t be a discrete time. Let $\boldsymbol{y}$ be a set of points in some area: $\boldsymbol{y} \equiv \{y_1, \ldots, y_p\}$, $y_i \in \boldsymbol{R}^2$, where p denotes the number of points. Note that $y_i \in \boldsymbol{R}^2$ corresponds to positions on the ground in some examples such as communication terminals and power grid. Let a pair of points $(x_1(t), x_2(t))$, $x_1(t) \neq x_2(t)$, be an input at time t. The input is selected randomly from the set $\boldsymbol{y}$ at time t. The growing network is represented by nodes $n_i(t) \in \boldsymbol{R}^2$ and edges $e_{ij}(t)$ between nodes $n_i(t)$ and $n_j(t)$ where $e_{ij}(t) = 1$ and $e_{ij}(t) = 0$ mean connection and disconnection, respectively. Let $\boldsymbol{n}(t)$ be a set of nodes: $\boldsymbol{n}(t) \equiv \{n_1(t), \ldots, n_{N(t)}(t)\}$ where $N(t)$ denotes the number of nodes at time t. In the algorithm, $N(t)$ can increase and the network can grow. The i-th node $n_i(t)$, $i \in \{1, \ldots, N(t)\}$, is a 2-dimensional vector corresponding to a synaptic vector of GCS. In order to memorize learning history we prepare a signal counter $SC_i(t)$ for node $n_i(t)$.

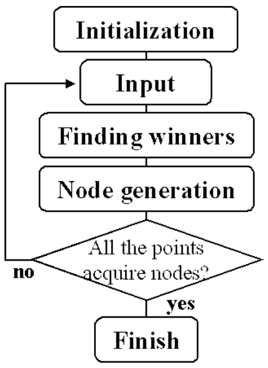

Figure 2: The flow chart of learning algorithm.

Our algorithm consists of the following 5 steps (see Fig. 2).

Step 1 (Initialization)
Let $t = 0$ and let $N(t) = 2$. At $t = 0$ two nodes $n_1(t)$ and $n_2(t)$ are located randomly and are connected to each other, $e_{12}(t) = 1$. Initial value of counters are $SC_1(t)$= $SC_2(t) = 0$.

Step 2 (Input)
We select a pair of points randomly from the set $\boldsymbol{y}$ and apply it as input $(x_1(t), x_2(t))$.

Step 3 (Finding winners)
We find two winner nodes $n_{c1}(t)$ and $n_{c2}(t)$ that are the closest to $x_1(t)$ and $x_2(t)$, respectively.

$$||x_1(t) - n_{c1}(t)|| = \min_i ||x_1(t) - n_i(t)|| \tag{1}$$

$$||x_2(t) - n_{c2}(t)|| = \min_i ||x_2(t) - n_i(t)|| \tag{2}$$

where $n_{c1}(t) \neq n_{c2}(t)$. We find the winner node $n_{c1}(t)$ at first and then find $n_{c2}(t)$. If there exist plural closest nodes for $x_1(t)$ or $x_2(t)$, *we declare the node with smaller subscription i as the winner*. Two winner nodes move to the input position and the signal counters $SC_{c1}(t)$ and $SC_{c2}(t)$ are updated.

$$n_i(t+1) = \begin{cases} x_i(t) & \text{if } i \in \{c_1, c_2\} \\ n_i(t) & \text{otherwise} \end{cases}$$
$$SC_i(t+1) = \begin{cases} SC_i(t) + 1 & \text{if } i \in \{c_1, c_2\} \\ SC_i(t) & \text{otherwise} \end{cases} \tag{3}$$

Step 4 (Node generation)
We refer to a node $n_i(t)$ as a fire-node if the signal counter reaches a threshold th. The threshold th is the first parameter of this algorithm to determine whether new nodes can generate or not. A new node $n_{N(t)+1}(t)$ is generated at the same position as the fire-node $n_i(t)$ and have connects to the fire-node and its closest neighbor node $n_j(t)$ as shown in Fig. 3.

The counter value of the new node is shared with the fire-node.

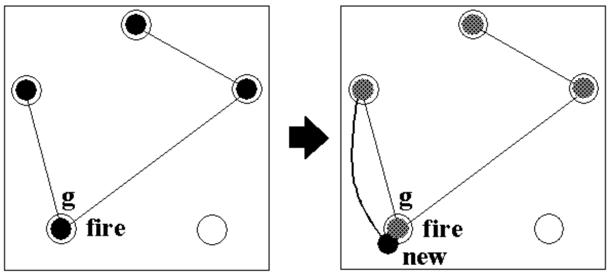

Figure 3: Node generation: a new node is generated at position g and have connects to the fire-node and its closest neighbor node.

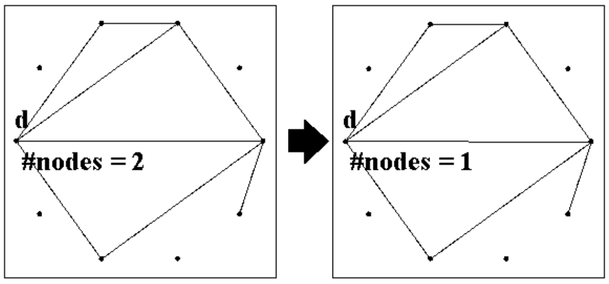

Figure 4: Node deletion: one of the two nodes at position d and its edge is deleted.

$$n_{N(t)+1}(t) = n_i(t) \quad \text{for } SC_i(t) = th \tag{4}$$

$$SC_{N(t)+1}(t) = SC_i(t) = \frac{th}{2} \tag{5}$$

$$e_{(N(t)+1)i}(t) = 1, \; e_{(N(t)+1)j}(t) = 1 \tag{6}$$

If one node fires then $N(t) = N(t)+1$. There exists possibility of two fire-nodes and $N(t) = N(t) + 2$. *If plural node can exist at the same position, older node (having smaller subscription) can move to the other position for networking by Equation (3). We then introduce lifetime M to suppress unnecessary generation of new nodes*: we delete a new node as the following if it exists with other node(s) during M time steps as shown in Fig. 4 .

$$\{n_{i+1}(t), \dots, n_{N(t)}(t)\} \rightarrow \{n_i(t), \dots, n_{N(t)-1}(t)\} \tag{7}$$

The lifetime M is the second parameter and the deletion means $N(t) = N(t)-1$. If there are two nodes to be deleted, then $N(t) = N(t) - 2$.

Step 5 (Termination)
If all the points acquire nodes: $\boldsymbol{n}(t) \cap \boldsymbol{y} = \boldsymbol{y}(t)$, then learning is terminated. Otherwise go to Step 2 with $t = t + 1$ and $N(t + 1) = N(t)$.

3. Numerical Experiments

In order to evaluate network structure we introduce two basic measures. The first one is the average path length

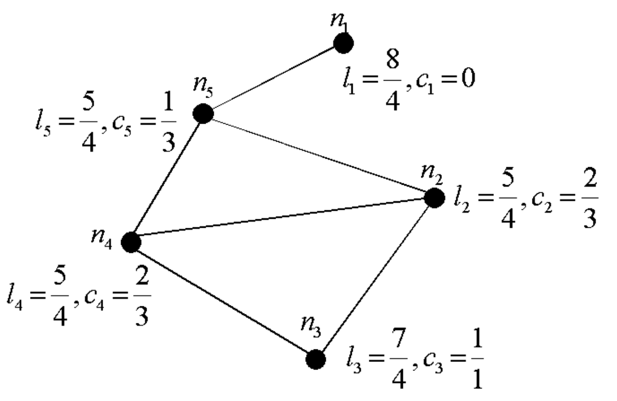
n_1
$l_1=\frac{8}{4}, c_1=0$
n_5
$l_5=\frac{5}{4}, c_5=\frac{1}{3}$
n_2
$l_2=\frac{5}{4}, c_2=\frac{2}{3}$
n_4
$l_4=\frac{5}{4}, c_4=\frac{2}{3}$
n_3
$l_3=\frac{7}{4}, c_3=\frac{1}{1}$

Figure 5: Path length L and clustering coefficient C. In this example, $s_{12}=2$, $s_{13}=3$, $s_{14}=2$, $s_{15}=1$, $l_1=\frac{8}{4}$, $s_{21}=2$, $s_{23}=1$, $s_{24}=1$, $s_{25}=1$, $l_2=\frac{5}{4}$ and $L=\frac{30}{20}$; $k_1=1$, $E_1=0$, $c_1=\frac{0}{1}$, $k_2=3$, $E_2=2$, $c_2=\frac{2}{3}$ and $C \doteq 0.53$.

L that measures typical separation the between two nodes. Let s_{ij} be the shortest path length between nodes i and j and let l_i be the average shortest path length for node i. Averaging l_i for all the nodes, we obtain the average path length.

$$L=\frac{1}{N}\sum_{i}^{N} l_i, \quad l_i=\frac{1}{N-1}\sum_{i\neq j}^{N} s_{ij} \tag{8}$$

where N is the number of nodes after the learning. An example is shown in Fig. 5.

The second one is the clustering coefficient C that measures cliquishness of a typical neighborhood. For i-th node n_i, let k_i be the number of nodes connecting with n_i, E_i be the number of existing edges among the k_i nodes and let c_i be the ratio of E_i per the number of possible edges $\frac{1}{2}k_i(k_i-1)$. Averaging c_i for all the nodes, we obtain the clustering coefficient.

$$C=\frac{1}{N}\sum_{i}^{N} c_i, \quad c_i=\frac{E_i}{\frac{1}{2}k_i(k_i-1)} \tag{9}$$

An example is shown in Fig. 5.

We have performed two basic numerical experiments. In the experiments the two parameters are fixed by trial-and-errors:

$$M=3, \quad th=10 \tag{10}$$

3.1. Experiment 1 (circular input space)

First we consider 50 points located on a circle as shown in Fig. 6. In the experiment, $\boldsymbol{y}$ consists of 50 nodes on the circumference with center $(0,0)$: $\boldsymbol{y}\equiv\{y_1,\ldots,y_{50}\}$, $y_i\in\boldsymbol{R}^2$ and $|y_i|=1$. The input of a pair $(x_1(t),x_2(t))$ is selected randomly from set $\boldsymbol{y}$ and are applied to the network. Fig. 6 shows the learning process with growing cells structure. In the beginning, the nodes tend to have long edge to the other nodes. We refer to a node having a long edge as a key node. It should be noted that an older node with smaller subscription tends to be the key node because the older node has priority to be winner thus to have large signal counter value in Step 3. As the network grows, key nodes tend to have short edge and SWN can be obtained as confirmed in Tables 1 and 2: we have obtained relatively small L and large C for $\sharp points \in \{40,50,60\}$. The column ALG WS shows results by WS algorithm in [9] with reconnection probability $P=0.07$. We select suitable value in [9].

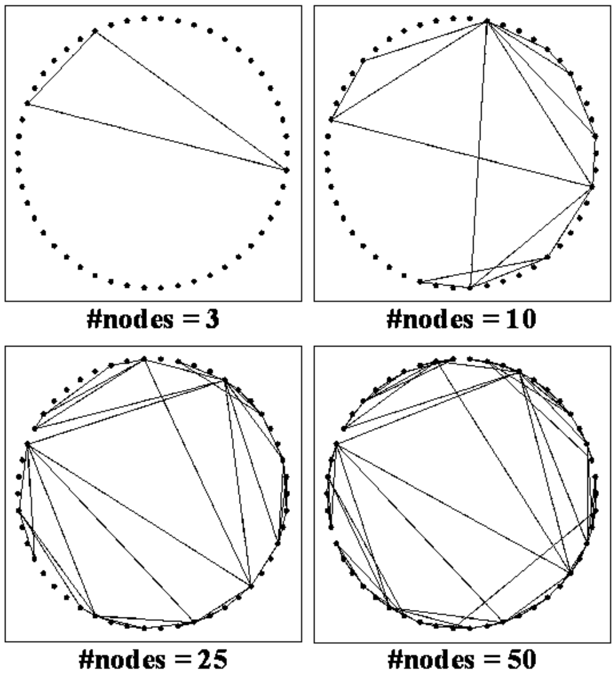

Figure 6: Learning process for circular input space. ($\sharp$points = 50)

3.2. Experiment 2 (input space of towns)

Next we apply the algorithm to a practical input space: 50 towns in Berlin. Fig. 7 shows the learning process with growing cells structure. In this case, we have obtained SWNs characterized by relatively small $L \doteq 1.92$ and large $C \doteq 0.62$.

Table 1: Average path length L for circular input space.

♯points	ALG proposed	ALG WS ($P=0.07$)
40	1.90	4.22
50	1.92	4.51
60	1.93	5.41

Table 2: Cluster coefficient C for circular input space.

♯points	ALG Proposed	ALG WS ($P = 0.07$)
40	0.57	0.47
50	0.58	0.43
60	0.63	0.45

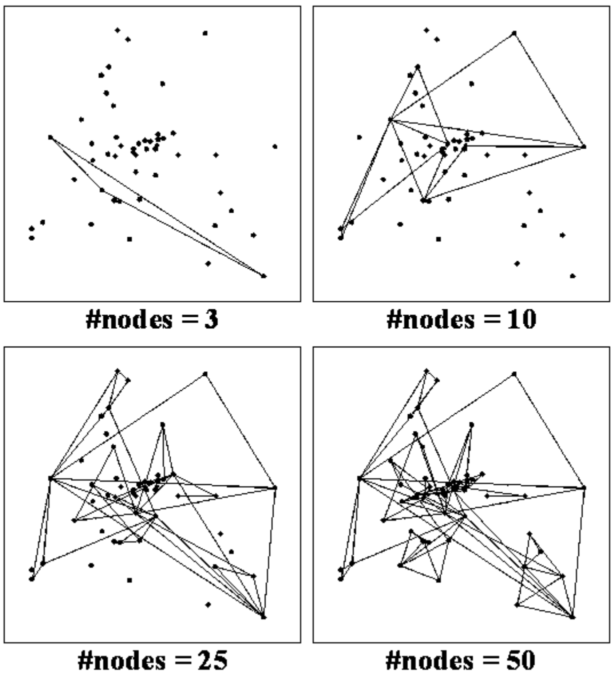

Figure 7: Learning process for input space of 50 towns in Berlin. Two measures after the learning are $L \doteq 1.92$ and $C \doteq 0.62$.

4. Conclusions

We have studied a flexible learning algorithm for networking based on growing SOM. Performing the basic numerical experiments, we have confirmed that our algorithm can generate small-world like networks and the network structure as parameters are selected suitably. Future problems include the following:
(1) Analysis of roles of learning parameters.
(2) Statistical analysis of the learning process and resulting network structure.
(3) Application to practical data such as power grid and communication network.

Acknowledgments

The authors would like to thank Dr. Hiroyuki Torikai and Mr. Hiroki Sasamura for their help.

References

[1] T. Kohonen, Self-organization and associative memory, 2nd Ed. Springer-Verlag, Berlin, 1988.

[2] B. Fritzke, Growing cell structures - a self-organizing network for unsupervised and supervised learning, Neural Networks, 7, pp. 1441-1460, 1994.

[3] S. Kawahara and T. Saito, An adaptive self-organizing algorithm with virtual connection, Journal of Advanced Computational Intelligence, pp. 203-207, 1997.

[4] B. Angeniol, G. de La C. Vaubois and J. Y. Le Texierr, Self-organizing feature maps and the traveling salesman problem, Neural Networks, 1, pp. 289-293, 1988.

[5] H. Sasamura, R. Ohta and T. Saito, A simple learning algorithm for growing ring SOM and its application to TSP, Proc. of ICONIP, cr1508, 2002.

[6] T. Ehara, H. Sasamura and T. Saito, An approach to the TSP based on growing self-organizing maps, Proc. NOLTA, pp. 709-712, 2004.

[7] R. Singh, V. Cherkassky and N. Papanikolopoulos, Self-Organizing Maps for the Skeletonization of Sparse Shapes, IEEE Trans. Neural Networks, 11, 1, pp. 241-248, 2000.

[8] H. Sasamura and T. Saito, A Simple Learning Algorithm for Growing Self-Organizing Maps and Its Application to the Skeletonization, Proc. of IJCNN, pp. 787-790, 2003.

[9] D. J. Watts and S. H. Strogatz, Collective Dynamics of 'Small-World' Networks, Nature, 393, pp. 440-442, 1998.

[10] X. F. Wang and G. Chen, Complex Networks: Small-World, Scale-Free and Beyond, IEEE Circuits and Systems Magazine, pp. 6-20, 2003.

[11] L. F. Lago-Fernandez, R. Huerta, F. Corbacho, J. A. Siguenza, Fast response and temporal coherent oscillations in small-world networks, Phys. Lett. 84, 12, pp.2758-2761, 2000.

[12] H. Sasamura, T. Saito and R. Ohta, A simple learning algorithm for network formation based on growing self-organizing maps, IEICE Trans. Fundamentals, E87-A, 10, pp.2807-2810, 2004.