Unsupervised Neural Network based Topological Learning from Point Clouds for Map Building

Yuichiro Toda, Weihong Chin, Naoyuki Kubota
Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo
Japan

Abstract:
Topological structure learning methods are expected for the field of data mining for extracting multiscale topological structures from an unknown dataset. In this paper, we introduce the unsupervised neural network method for topological structure learning method from point clouds for map building. We propose Batch Learning GNG (BL-GNG) in order to improve the learning convergence. BL-GNG uses an objective function based on Fuzzy C-means for improving the learning convergence. Finally, we conduct on several experiments for evaluating our proposed method by comparing to other hierarchical approaches, and discuss the effectiveness of our proposed method.

1. Introduction

Recently, various types of robots have emerged in many fields as a progress of robot technologies. Especially, the expectation of disaster robots, which can be robustly utilized in a disaster area, is increasing for preventing the second disaster in the area [1]. It is important to extract the environmental information related to a movable area of the robot and a dangerous area such as rubble with the high possibility of collapse in order to act safely and quickly in the disaster area. In this paper, we focus on an environmental sensing and environmental map building technology using a 2D/3D point cloud for extracting the efficient and effective information. Specifically, we introduce a topological structure learning method from the point clouds because the topological structure of the point clouds is useful for the environmental map building methods by utilizing the topologies. In order to learn the topological structure of the point clouds, many kinds of unsupervised learning methods are proposed by many researchers. The objectives of unsupervised learning are feature extraction, clustering and learning the topological structure of a dataset to find important information efficiently [2]. In this paper, we explain Growing Neural Gas (GNG) based geometrical feature extraction method for building the environmental map. GNG proposed by Fritzke is one of the unsupervised neural networks [3]. The unsupervised neural network is performed by using only data without any teaching signals. Self-organized map (SOM), neural gas (NG), growing cell structures (GCS), and GNG are well known as unsupervised neural networks [4-6]. Basically, these methods use the competitive learning. The number of nodes and the topological structure of the network in SOM are designed beforehand. In NG, the number of nodes is fixed beforehand, but the topological structure is updated according to the distribution of sample data. On the other hand, GCS and GNG can dynamically change the topological structure based on the adjacent relation (edge) referring to the ignition frequency of the adjacent node according to the error index. However, GCS does not delete nodes and edges, while GNG can delete nodes and edges based on the concept of ages. Furthermore, GCS must consist of k -dimensional simplexes whereby k is a positive integer chosen in advance. The initial configuration of each network is a k -dimensional simplex, e.g., a line is used for k=1, a triangle for k=2, and a tetrahedron for k=3. GCS has applied to construct 3D surface models by triangulation based on 2-dimensional simplex. However, because the GCS does not delete nodes and edges, the number of nodes and edges is over increasing. Furthermore, GCS cannot divide the sample data into several segments. On the other hand, GNG also has problems related to learning convergence because GNG is on-line learning method and the learning rates do not depend on the number of iteration. Therefore, we propose Batch Learning GNG (BL-GNG) in order to improve the learning convergence and reduce the user-designed parameters. The learning method of BL-GNG is based on Fuzzy C-means (FCM) because FCM has a local minimum convergence property [7, 8].

This paper is organized as follows. Section 2 proposes BL-GNG. Section 4 shows several experimental results. Finally, we summarize this paper.

2. Growing Neural Gas based Approach

In this chapter, we propose Batch Learning GNG (BL-GNG). GNG is on-line learning and the learning rates do not depend on the number of iterations. Therefore, GNG has a problem related to learning stability. For improving learning stability, we change the on-line learning to a batch learning from the point of view Fuzzy C-means (FCM).

2.1 Update adjacent relation

It is important not to depend on the input order for realizing the batch learning algorithm. In GNG, the update rules of edge and node strongly depend on the input order.

Therefore, we explain the update rule of adjacent relation in BL-GNG.

In the update rule, BL-GNG uses a temporary adjacent relation $g_{i,j}$ of a temporary adjacency matrix G in order to create and remove the edge. At the initial step, all the temporary adjacent relations are set to 0. When the input vector v_i is given, the temporary adjacent relation is updated as follows;

$$\begin{aligned} & g_{s_1,s_2} = 1 \\ & s_1 = \arg\min_{j \in W} \left\| v_i - w_j \right\|, \;\; s_2 = \arg\min_{j \in W \backslash s1} \left\| v_i - w_j \right\| . \end{aligned} \tag{1}$$

After all the input vectors are given, all the elements $c_{i,j}$ of adjacency matrix are updated as follows;

$$\begin{aligned} & c_{i,j} = 1 \quad \text{if } g_{i,j} = 1 \\ & c_{i,j} = 0 \quad \text{otherwise} \end{aligned} . \tag{2}$$

In Batch learning, the winner node s_1 and the second winner s_2 node are calculated for all input vectors. The temporary adjacent relation that is $g_{i,j}$= 0 is not selected as s_1 and s_2. Therefore, we can remove such an edge $c_{i,j}$ because there is no input vector between node w_i and w_j. In this way, BL-GNG updates the adjacent relation.

```
Algorithm 1 Initialization:
Input: Input dataset $V$, Number of initial nodes $M_{\text{init}}$
Output: Set of node $W$ and Adjacency Matrix $C$
for $j$=1 to $M_{\text{init}}$ do
    Select the $k$th input vector randomly
    $w_j = v_k$
end for
for $j$=1 to $N$ do
    $s_1 = \arg\min_{i \in A} \| v_j - w_i \|$
    $s_2 = \arg\min_{i \in A \backslash \{s_1\}} \| v_j - w_i \|$
    $a_{s1,s2}$=1
end for
for $i$=1 to $M_{\text{init}}$ do
    for $j$=1 to $M_{\text{init}}$ do
        if $g_{i,j}$=1 then
            $c_{i,j}$=1
        else
            $c_{i,j}$=0
        end if
    end for
end for
```

2.2 Initialization of BL-GNG

In GNG, the number of initial nodes is 2. In BL-GNG, it takes a long time to grow the number of required nodes because the insertion of a new node performs after all input vectors are calculated. However, BL-GNG can use any number of the initial nodes by using the update rule of adjacent relation. Algorithm 1 shows the detail procedure. In Initialization, the adjacency matrix C sets to zero and M_{init} nodes are generated by selecting M_{init} input vector randomly. After assigning all nodes, the update rule enables initialization of BL-GNG to start from any number of nodes.

2.3 Learning rule of BL-GNG

The update rule of BL-GNG is based on FCM learning method. Therefore, the update rule of the ith node w_i is calculated by the following equation;

$$w_i^{(t)} = \frac{\sum_{j=1}^{N} \left(u_{ji}^2 \cdot v_j \right)}{\sum_{j=1}^{N} u_{ji}^2} . \tag{3}$$

The membership function of BL-GNG must satisfy the following conditions.

$$\begin{aligned} & u_{nk} \in [0,1]; \quad 1 \le n \le N; \quad 1 \le k \le M; \\ & \sum_{i=1}^{M} u_{ni} = 1; \qquad 1 \le n \le N; \\ & \sum_{j=1}^{N} u_{jk} > 0; \qquad 1 \le k \le M; \end{aligned} \tag{4}$$

From these conditions, the membership function of BL-GNG is defined as follows;

$$u_{ji} = \frac{1}{\left(\dfrac{d_{i,j}}{d_{s_1,j}} + \sum_{c_{k,s1}=1} \dfrac{d_{i,j}}{d_{k,j}} \right)} . \tag{5}$$

Equation (5) indicates that the membership function influences only the winner node and the nodes with adjacent relation to the winner node. Generally, FCM has a fuzziness m that is a user-defined parameter for controlling the influence the nodes for the input vector. BL-GNG uses the fixed fuzziness value m = 2 since BL-GNG can control the influence for the nodes by changing the adjacent relations dynamically. In addition, equation (5) sometimes generates the nodes that do not satisfy the condition (4). Such nodes are removed from the set of nodes W. This procedure is similar to a procedure of GNG-U [9] that is used for non-stationary dataset. GNG-U removes the nodes whose utility value is lower than a user-defined threshold value.

2.4 Total procedure of BL-GNG

We explain the total algorithm of BL-GNG. BL-GNG can be divided two procedures. One is calculating the membership function and the temporary adjacent relations. Algorithm 2 shows the algorithm of a local learning, where $H^1 = \{h^1_1, h^1_2, \ldots, h^1_M\}$ and $H^2 = \{h^2_1, h^2_2, \ldots, h^2_M\}$ are the variables for calculating the positions of nodes by using equation(10). In Algorithm 2, the accumulated error E_i is calculated for using a criterion in an insertion algorithm of new node. After all input vectors are calculated, all nodes and the adjacency matrix are updated (Algorithm 3).

Algorithm 4 shows the insertion algorithm of new node and Algorithm 5 shows the total procedure of BL-GNG. Algorithm 4 is basically the same as GNG. However, BL-GNG does not need to decrease the accumulated errors because the accumulated errors set to 0 before Algorithm 2. In addition, BL-GNG does not need a parameter λ that is used as a criterion of insertion algorithm because all input vectors are calculated in Algorithm 2. Therefore, the insertion algorithm is performed after Algorithm 3. In this way, BL-GNG learns the topological structure.

Table 1 shows the feature of BL-GNG compared to GNG. In Table 1, the convergence means the amount of update nodes as the iterations progress. In GNG, the amount does not converge since the learning process strongly depends on the input order and the learning rate does not depend on the number of iteration. Therefore, the convergence is low in Table 1. Next, we consider the number of the user-defined parameters. GNG requires seven parameters as the learning rate, the maximum age of edge $a_{\max}$, integer multiplier λ, 2 decrease rates of the accumulated error α, β and the maximum number of nodes $M_{\max}$. On the other hand, BL-GNG requires only two user-defined parameters as the number of initial nodes M_{init} and the maximum number of nodes. Furthermore, we generally use M_{init} = 2 or M_{init} = $M_{\max}$ unless BL-GNG is used specially. In such a case, the number of user-designed parameters is only 1. In this way, BL-GNG has several advantages compared to GNG.

```
Algorithm 2 Local learning:
Input: Set of input data $V$, set of nodes W and
       adjacency matrix $C$
Output: $H^1$, $H^2$, temporary adjacency matrix $G$ and
        the accumulated error $E$
Set all $h^1_j$=0 and $h^2_j$=0, $g_{i,j}$=0, $E_i = 0$
for $j$=1 to $N$ do
    $s_1 = \arg\min_{i \in A} \|v_j - w_i\|$
    $s_2 = \arg\min_{i \in A \setminus \{s_1\}} \|v_j - w_i\|$
    Calculate the membership functions and update the
      variables $h^1_i$ and $h^2_i$ of the winner node s1 and
      the neighborhood of the s1
        $h^1_i \leftarrow h^1_i + u^2_{ji}$
        $h^2_i \leftarrow h^2_i + u^2_{ji} \cdot v_j$
    $g_{s1,s2}$=1
    Update the accumulated error $E_{s1}$
        $E_{s_1} \leftarrow E_{s_1} + \|v_j - w_{s_1}\|^2$
end for
```

```
Algorithm 3 Updating procedure:
Input: $H^1$, $H^2$ and temporary adjacency matrix $G$
Output: Set of nodes $W$ and adjacency matrix $C$
for $i$=1 to $M$ do
    Update the node $w_i$ as following;
        $w_i^{(t)} = h_i^2 / h_i^1$
    for $j$=1 to $M$ do
        Update the edge $c_{i,j}$ by calculating Eq. (2)
    end for
end for
```

```
Algorithm 4 Insertion algorithm of a new node (IA):
Input: Set of accumulated error $E$, set of nodes $W$ and
       adjacency matrix $C$
Output: Set of nodes
Select the unit $q$ with the maximal accumulated error
Select the unit $f$ with the maximal accumulated error
  among the neighbors of $q$
Insert a new unit $r$ halfway between $q$ and $f$
    $w_r = 0.5 \cdot (w_q + w_f)$
Insert edges $c_{q,r}$=1, $c_{r,f}$=1 and remove the edge
  $c_{q,f} = 0$
```

```
Algorithm 5 BL-GNG:
Input: Set of input data $V$, the number of initial nodes
       $M_{\text{init}}$ and the maximum number of nodes $M_{\max}$
Output: Set of nodes $W$ and Adjacency Matrix $C$
$M = M_{\text{init}}$
Initialization ($V$)
while the stopping criterion is not reached do
    ($H^1$, $H^2$, $G$, $E$) = Local learning ($V$, $W$, $C$)
    ($W'$, $C$) = Update procedure ($H^1$, $H^2$, $G$)
    if $M \le M_{\max}$ do
        ($W$, $C$) = IA ($E$, $W$, $C$)
        $M \leftarrow M + 1$
    end if
end while
```

TABLE 1 FEATURE OF BL-GNG COMPARED TO GNG

	GNG	BL-GNG
Initial node number	2	User-defined
Convergence	Low	High
Computational cost	Constant	Depending on number of data
Required memory	Small	Large
Parameters	7	2

3. EXPERIMENTAL RESULT

In this section, we performed an experiment to show the learning capability of BL-GNG compared with GNG. In this experiment, we used a 2D plot dataset that is composed of 3 circles (Fig.1) and the centers and radiuses of the circles are

C1 = (x, y, r) = (6.0, 6.0, 0.5), C2 = (6.0, 8.0, 0,5) and C3 = (9.0, 7.0, 1.5), respectively. The plot data is generated by a uniform random number into each circle. The numbers of data of C1, C2 and C3 are 200, 200 and 300, respectively. In BL-GNG, we defined M_{init} = 2 and $M_{\max}$ = 100 as the parameters. In GNG, we defined η_1= 0.08, η_2= 0.008, $a_{\max}$= 200, α=0.5, β=0.995, λ=1200. The integer multiplier λ and the maximum age of the edge $a_{\max}$ were defined for the same conditions of BL-GNG and the other parameters are defined empirically. In this experiment, the number of trials is 100 in each method. To evaluate the convergence of each method, we used the following evaluation criterion;

$$U = \sum_{k=1}^{M} \left\| \Delta w_k \right\|^2 \Big/ M, \qquad (13)$$

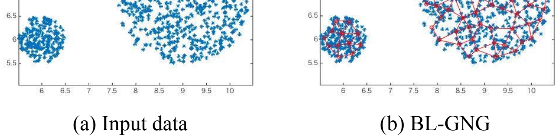
$$\Delta w_k = w_k^t - w_k^{(t-1)}$$

where w^t_k is the position of the kth nodes at the tth iteration and U means the average update amount of nodes. Therefore, the criterion is low as the learning grows if the learning method is stable. In addition, we used the objective function of k-means method to evaluate the learning result of the topological structure accurately.

$$E = \sum_{n=1}^{N} \sum_{k=1}^{M} r_{nk} \left\| v_n - w_k \right\|^2$$

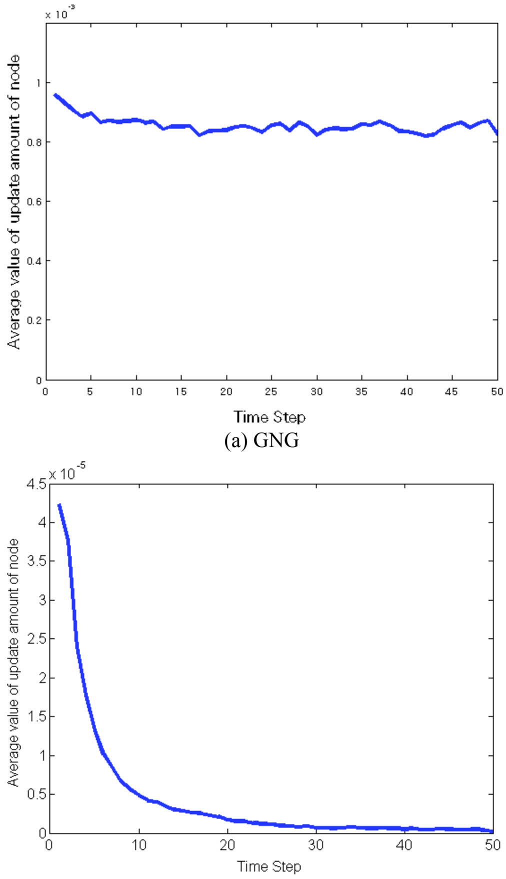
$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \left\| v_n - w_j \right\|^2 \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

Fig. 2 plots the average update amount after the number of nodes M reaches the maximum number of nodes $M_{\max}$. In Fig. 2 (a), GNG did not converge because the average update amount of GNG had constant value after M reaches $M_{\max}$. This result shows that the topological learning of GNG depends on the order of input vectors since the learning rate of GNG does not depend on the number of iteration. On the other hand, BL-GNG converged to 0 (Fig. 2 (b)) though the membership function of BL-GNG also dose not depend on the number of iteration. Because we consider that BL-GNG has the local minimum convergence property by calculating the node positions from all input vectors. Furthermore, the error of BL-GNG is almost the same as that of GNG (Table 2). From these results, BL-GNG improves the learning stability of GNG as the learning accuracy of GNG remains.



(a) Input data (b) BL-GNG

Fig. 1. An example of a topological structure of BL-GNG with M_{init} = 2 and $M_{\max}$ = 100. Blue and red dots indicate input data and node, respectively. Red line indicates edge.



(a) GNG

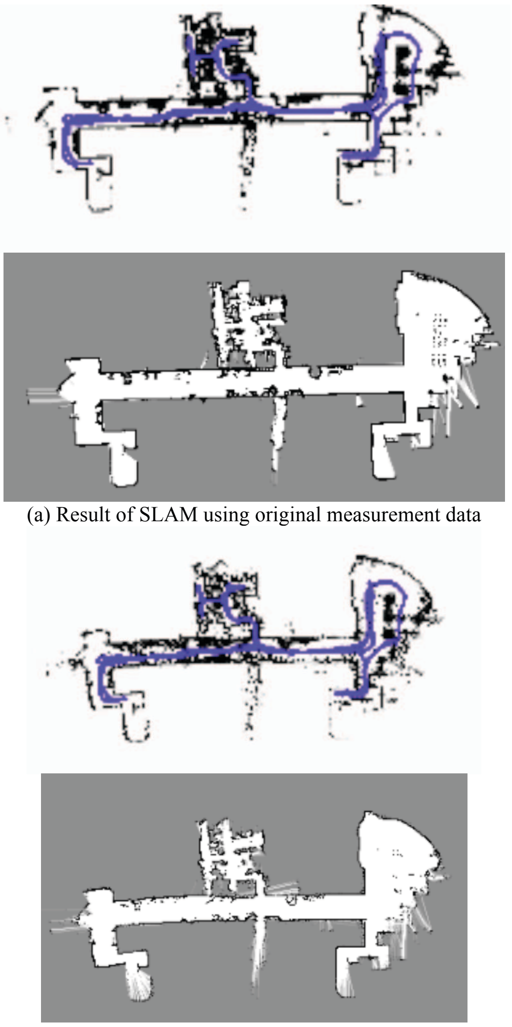
(b) BL-GNG

Fig. 2 Experimental results of average update value of the nodes. Time step is the number of iteration after $M_{\max}$ nodes are generated.

TABLE II EXPERIMENTAL RESULT OF ERROR VALUE

	BL-GNG	GNG
Average	14.10	15.27
Variance	0.012	0.137

Next, we conduct on an experiment about the 2D environmental map building using the measurement data from laser range finder (LRF). The number of nodes is 3 percent of the number of measurement data at each measurement for verifying that BL-GNG can extract the geometrical feature from the point clouds effectively, and the average numbers of the original measurement data and nodes are 830.7 and 21.5, respectively. In this experiment, we use the evolution strategy based localization and map building method proposed by [10]. Fig.3 shows the experimental results of map building. While Fig. 3 (b) has some distortions compared with Fig.3 (a), there is a degree of accuracy in the localization result because the map shape is almost the same between result of using the original measurement data and the BL-GNG nodes. In this way, our proposed method can extract the geometrical feature from the point clouds.



(a) Result of SLAM using original measurement data

(c) Result of SLAM using the BL-GNG nodes as the input data

Fig. 3 Results of SLAM in corridor environment.

4. Conclusion

In this paper, we introduced the unsupervised neural network for learning topological structure from the point clouds, and proposed Batch Learning Growing Neural Gas (BL-GNG) for improving the learning stability of GNG. The objective function of BL-GNG uses FCM type objective function because this type of the objective function has the local convergence property. We demonstrated that BL-GNG has the local convergence property comparing to GNG and applied BL-GNG to the localization and map building problem using real measurement data. We will apply the real-world applications such as analyzing a 3D point cloud for verifying the effectiveness of BL-GNG as the future work.

Acknowledgement

This work was funded by ImPACT Program of the Council for Science, Technology and Innovation.

References

[1] Murphy, Robin R., et al. "Search and rescue robotics." Springer Handbook of Robotics. Springer Berlin Heidelberg, 2008. 1151-1173.

[2] Richard O. Duda, Peter E. Hart and David G. Stork, “Pattern Classification,” 2nd ed, John Wiley & Sons, 2012.

[3] B. Fritzke, “A growing neural gas network learns topologies,” in Advances in Neural Information Processing Systems 7. MIT Press, pp. 625–632, 1995.

[4] Kohonen, Teuvo, “Self-organizing maps.” Vol. 30, Springer Science & Business Media, 2001.

[5] Martinetz, T., & Schulten, K., “A" neural-gas" network learns topologies.” , pp. 397-402, University of Illinois at Urbana-Champaign, 1991.

[6] B. Fritzke. "Growing cell structures—a self-organizing network for unsupervised and supervised learning." Neural networks vol.7, no.9, pp.1441-1460, 1994.

[7] Bezdek, J.C., "A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.PAMI-2, no.1, pp.1,8, Jan. 1980.

[8] Hathaway, Richard J., and James C. Bezdek. "Recent convergence results for the fuzzy c-means clustering algorithms." Journal of Classification 5.2 (1988): 237-247.

[9] Fritzke, Bernd, "A self-organizing network that can follow non-stationary distributions." Artificial Neural Networks—ICANN'97, Springer Berlin Heidelberg, pp.613-618, 1997.

[10] Yuichiro Toda, and Naiyuki Kubota. "Self-localization Based on Multi-resolution Map for Remote Control of Multiple Mobile Robots", Industrial Informatics, IEEE Transactions on, vol. 9, no. 3, pp. 1772-1781, 2013.

Unsupervised Neural Network based Topological Learning from Point Clouds for Map Building

Yuichiro Toda, Weihong Chin, Naoyuki Kubota
Tokyo Metropolitan University

Recently, various types of robots have emerged in many fields as a progress of robot technologies. Especially, the expectation of disaster robots, which can be robustly utilized in a disaster area, is increasing for preventing the second disaster in the area. It is important to extract the environmental information related to a movable area of the robot and a dangerous area such as rubble with the high possibility of collapse in order to act safely and quickly in the disaster area. In this paper, we focus on an environmental sensing and environmental map building technology using a 2D/3D point cloud for extracting the efficient and effective information. In this paper, we introduce the unsupervised neural network method for topological structure learning method from point clouds for the map building, and propose Batch Learning GNG (BL-GNG) in order to improve the learning convergence. BL-GNG uses an objective function based on Fuzzy C-means for improving the learning convergence. Finally, we conduct on several experiments for evaluating our proposed method by comparing to other hierarchical approaches, and discuss the effectiveness of our proposed method.