Growing Neural Gas with Correntropy Induced Metric

Naoki Masuyama
Faculty of Computer Science and Information Technology,
University of Malaya,
50603 Kuala Lumpur, MALAYSIA
Email: naoki.masuyama17@um.edu.my

Chu Kiong Loo
Faculty of Computer Science and Information Technology,
University of Malaya,
50603 Kuala Lumpur, MALAYSIA
Email: ckloo.um@um.edu.my

***Abstract*—This paper discusses the Correntropy Induced Metric (CIM) based Growing Neural Gas (GNG) architecture. CIM is a kernel method based similarity measurement from the information theoretic learning perspective, which quantifies the similarity between probability distributions of input and reference vectors. We apply CIM to find a maximum error region and node insert criterion, instead of euclidean distance based function in original GNG. Furthermore, we introduce the two types of Gaussian kernel bandwidth adaptation methods for CIM. The simulation experiments in terms of the affect of kernel bandwidth σ in CIM, the self-organizing ability, and the quantitative comparison show that proposed model has the superior abilities than original GNG.**

***Index Terms*—Growing Neural Gas; Correntropy Induced Metric; Kernel Method; Clustering;**

I. Introduction

Clustering algorithms in the field of artificial neural networks have performed its usefulness in numerous research fields such as statistics, data mining and multivariate analysis. One of the typical clustering algorithms is Self-Organizing Map (SOM) [1] which is introduced by Kohonen. The original SOM has a fixed size network which is able to adapt to consecutive input data by changing its network topology. Conventionally, numerous studies based on SOM algorithm have introduced such as classification [2], [3], cluster analysis [4], [5] and vector quantization [6]. However, the fixed size network in SOM is limiting an applicability for further applications or a wider usage, such as the problems with large scale dynamic data. In general, Growing Neural Gas (GNG) [7] algorithm performs the superior ability to dynamic information due to its network growing ability. GNG has a self-adaptability to input data by increasing network size and its topological structure. The topological network in GNG is able to represent the input data with more flexible way than a fixed size network like SOM. Furthermore, it can be improving visualization capabilities and understandings of input data, simultaneously. Due to its usefulness and effectiveness, it has widely accepted to numerous applications, such as robotics [8], computer vision [9] and complex data set modeling [10], [11].

Conventionally, several types of studies have introduced to improve the learning algorithms for construction of topological network. Ghesmoune et al. [12] introduced a growing neural gas over data stream which allows us to discover clusters of arbitrary shapes without any assumptions on the number of clusters. Boonmee et al. [13] introduced a hybrid GNG by considering not only the distance between the clusters, but also the centroids of the each cluster to obtain more practical topology structure. Mohebi and Bagirov [14] applied an algorithm based on split and merge of clusters to initialize neurons. The initialization algorithm speeds up the learning process in large high-dimensional data sets. Palomo et al. [15] combined GNG and tree structure to represent a network topology, which is called Growing Neural Forest (GNF). GNF learns a set of trees so that each tree represents a connected cluster of data. The experimental results show that it outperforms some well-known foreground detectors both in quantitative and qualitative terms.

One of the successful approaches is to apply the kernel method for the network learning process [16]. Chalasani and Principe [17] discussed the kernel SOM in terms of a similarity measure called Correntropy Induced Metric (CIM) from the information theoretic learning perspective. Adapting the SOM in the CIM sense is equivalent to reducing the localized cross information potential, and information theoretic function that quantifies the similarity between two probability distributions based on Gaussian kernel function. In this paper, we introduce the CIM based similarity measurement to growing network architecture, called GNG-CIM, and also introduce the kernel bandwidth adaptation method for CIM. It can be expected that the proposed model shows a superior data dimension reduction ability than original GNG.

This paper is organized as follows; Section II briefly introduce the definition of CIM. Section III presents the details of GNG-CIM algorithm. Section IV describes simulation experiments to evaluate the abilities of proposed model. Concluding remarks are presented in Section V.

II. Definition of Correntropy Induced Metric

Correntropy is a generalized similarity measure between two arbitrary random data X and Y [18], which is defined as follows;

$$C_\sigma(X,Y) = E\left[\kappa_\sigma(X-Y)\right], \tag{1}$$

where, κ_σ is a kernel function that satisfies the Mercer's Theorem [19]. It induces a reproducing kernel Hilbert Space.

Therefore, it can be defined as the dot product of the two random variables in the feature space as follows;

$$C(X,Y) = E\left[\langle \phi(x) \rangle, \langle \phi(y) \rangle\right], \tag{2}$$

where, ϕ denotes a non-linear mapping from the input space to the feature space based on inner product operation as follows;

$$\kappa(x,y) = \left[\langle \phi(x) \rangle, \langle \phi(y) \rangle\right]. \tag{3}$$

In practical, correntropy can be described by the following equation due to the finite number of data L available;

$$\hat{C}_{L,\sigma} = \frac{1}{L}\sum_{i=1}^{L} \kappa_\sigma(x_i - y_i). \tag{4}$$

Here, correntropy is able to induce a metric, which is called Correntropy Indced Metric (CIM), in the data space. Let sample vectors $\boldsymbol{X} = [x_1, x_2, \ldots, x_L]$ and $\boldsymbol{Y} = [y_1, y_2, \ldots, y_L]$ are given, CIM can be defined as follows;

$$\begin{aligned} CIM(X,Y) &= \left[\kappa_\sigma(0) - \hat{C}_\sigma(X,Y)\right]^{\frac{1}{2}} \\ &= \left[\frac{1}{L}\sum_{i=1}^{L}\{\kappa_\sigma(0) - \kappa_\sigma(x_i - y_i)\}\right]^{\frac{1}{2}}. \end{aligned} \tag{5}$$

It can be considered that CIM in information theoretic learning perspective quantifies the similarity between two probability distributions.

III. Growing Neural Gas with Correntropy Induced Metric

Conventionally, Chalasani and Principe introduced CIM to Self-Organizing Map called SOM-CIM [17]. Although SOM-CIM showed the superior self-organizing ability, SOM-CIM has a certain limitation due to its fixed topological network. Thus, to overcome a drawback of SOM-CIM, we introduce correntropy to GNG, which is called GNG-CIM. As mentioned in Section II, CIM is calculated based on kernel function. Therefore, we also introduce the kernel adaptation method for GNG-CIM inspired by SOM-CIM. In the following subsections, we first present the fundamentals of GNG-CIM, then two types of kernel bandwidth adaptation methods are introduced; adjusted by (i) the number of nodes in topological network, and (ii) the distribution of nodes in topological network, respectively. In this paper, we utilize a Gaussian kernel for CIM, which is most popularly applied in the information theoretic learning.

A. Fundamentals of GNG-CIM

In regards to original GNG [7], the node which is most similar with input data, and the region which has a maximum error are defined by euclidean distance based calculations, respectively. On the other hand, the proposed model utilizes CIM based calculations. We consider that this is the significant difference between GNG-CIM and original GNG.

Let us suppose the input vector $\boldsymbol{V} = (v_1, v_2, \ldots, v_L)$ is given to the network at an instant l, the winner node is obtained using CIM as follows;

$$\begin{aligned} r &= \arg\min_k CIM(v(l), w)_{N=1} \\ &= \arg\min_k \left(\kappa_\sigma(0) - \kappa_\sigma\left(\|v(l) - w\|\right)\right), \end{aligned} \tag{6}$$

where, r denotes index of the node at instant l, k denotes the number of nodes in network, w_s denotes the reference vector, and σ denotes a kernel bandwidth.

Throughout the remaining sections, it is assumed that s denotes the index of 1st similar node, and t denotes the index of 2nd similar node, i.e., w_s and w_t denote 1st and 2nd similar reference vectors for all nodes k, respectively. Furthermore, the topological neighbors of s-th node (i.e., all nodes that have edge connection with s-th node) are indicated by e, such as topological neighbors reference vector w_e.

As mentioned earlier, the CIM is regarded as a local error between input and reference vectors for updating the weight, i.e.;

$$Err \leftarrow Err + \left[CIM(v(l), w)\right]^2. \tag{7}$$

The update rule for reference vector w is defined as follows;

$$w(l+1) = w(l) - \eta_w \Delta w, \tag{8}$$

where, η_w $(0 < \eta_w \leq 1)$ denotes learning rate. Here, the Gaussian kernel is utilized, then the gradient Δw is defined as follows;

$$\Delta w = \begin{cases} -G_\sigma\left(\|v(l) - w\|\right)(v(l) - w) & (r = s) \quad \text{(9a)} \\ -h_{rs} G_\sigma\left(\|v(l) - w\|\right)(v(l) - w) & \\ \qquad\qquad (r \neq s \in \text{neighbors } e), & \text{(9b)} \end{cases}$$

where, G_σ denotes a Gaussian kernel. h_{es} is given as follows;

$$h_{rs}(l) = \exp\left(-\frac{(w_r - w_s)^2}{2\sigma_h(l)}\right), \tag{10}$$

where, σ_h is the kernel bandwidth.

The update rules for network topology and node insert procedure are same as original GNG manner. In terms of original GNG, the new node will be inserted into the region that has maximum error based on euclidean distance. In contrast, GNG-CIM applies CIM based error calculation which is defined as Eq. (7). The entire process of GNG-CIM is presented in Algorithm 1.

B. Kernel Bandwidth Adaptation in GNG-CIM

In terms of kernel based clustering algorithm, to determine the appropriate kernel bandwidth σ is essential to control the performance of its model. In this paper, we introduce the two types of kernel bandwidth adaptation methods which are inspired by SOM-CIM [17]. Note that the following adaptive kernel algorithms are performed at step 6 in Algorithm 1.

Algorithm 1 Topological mapping in GNG-CIM

Require:
input vector $v(l)$
reference vector w
available maximum number of nodes N_{max}
maximum node age age_{max}
new node inserting cycle λ
error reduction rate for node inserting process α
error reduction rate for all nodes β
CIM criterion C_{cim}

Ensure: reference vector w

1: **Initialization**: Create two nodes and its reference vector with edge between two nodes, and setting zero values to CIM and edge age.
2: Input a vector $v(l)$ to a topological map
3: Calculate CIM and find 1st and 2nd similar reference vectors by Eq. (6)
4: Calculate the local error of s-th node by adding it to the squared CIM between reference vector w_s and input vector $v(l)$ by Eq. (7)
5: Update the reference vector w_s and its topological neighbor reference vectors w_e by Eqs. (8)-(10)
6: Kernel bandwidth adaptation
7: Increase the age of all edges outgoing from the s-th node
8: **if** s-th node and t-th node are connected, and $(\mathrm{CIM}_t - \mathrm{CIM}_s) \leq C_{cim}$ **then**
9: Set the age of their edge to zero
10: **else**
11: Create a new edge between s-th node and t-th node
12: **end if**
13: **for each** Edge age is larger than age_{max}, for all nodes **do**
14: Remove edge
15: **end for**
16: **for each** Node which has no more emanating edges **do**
17: Remove node
18: **end for**
19: Determine a p-th node which has a largest CIM
20: Determine a q-th node which has a largest CIM among neighbors of p-th node
21: **if** The number of current iterations is a multiple of λ, the number of nodes is smaller than N_{max}, and $CIM_q > C_{cim}$ **then**
22: Create a u-th node at the middle between p-th and q-th nodes as follows; $w_u = (w_p + w_q)/2$
23: Replace the edge between p-th and q-th nodes by the edges between p-th and u-th nodes, and q-th and u-th nodes, respectively
24: Decrease the errors of p-th and q-th nodes, and set the value of the error of u-th node as follows;
$Err_p \leftarrow \alpha \cdot Err_p$
$Err_q \leftarrow \alpha \cdot Err_q$
$Err_u \leftarrow Err_p$
25: Decrease errors of all nodes as follows;
$Err \leftarrow \beta \cdot Err$
26: **end if**
27: **if** $l < L$ **then**
28: Continue from step 2 with $l \leftarrow l + 1$
29: **end if**

1) Number of Nodes based Adaptation: This is one of a simplest methods to determine the kernel bandwidth which is controlled by the proportion of current number of nodes in topological network in available maximum number of nodes, such as;

$$\sigma_h(l) = \sigma_h^i \exp\left(-\theta \cdot \sigma_h^i \frac{N}{N_{max}}\right), \tag{11}$$

where, σ_h^i denotes initial value of σ, θ denotes decreasing rate of σ_h^i, N denotes a current number of nodes in topological network, and N_{max} denotes the available maximum number of nodes, respectively.

2) Node Distribution based Adaptation: Chalasani and Principe [17] extent the Kullback-Leibler divergence based density estimation algorithm [20] to the kernel bandwidth adaptation algorithm as follows; here, Kullback-Leibler divergence $D_{KL}(f||g)$ is estimated non-parametrically from the data by the Parzen window estimation [21]. If f is estimated Probability Density Function (PDF) of data $\boldsymbol{v}_l$ $(l \in 1, 2, \ldots, L)$ and g is estimated PDF of reverence vectors $\boldsymbol{w}_j$ $(j \in 1, 2, \ldots, N_{max})$, thus D_{KL} can be described as follows;

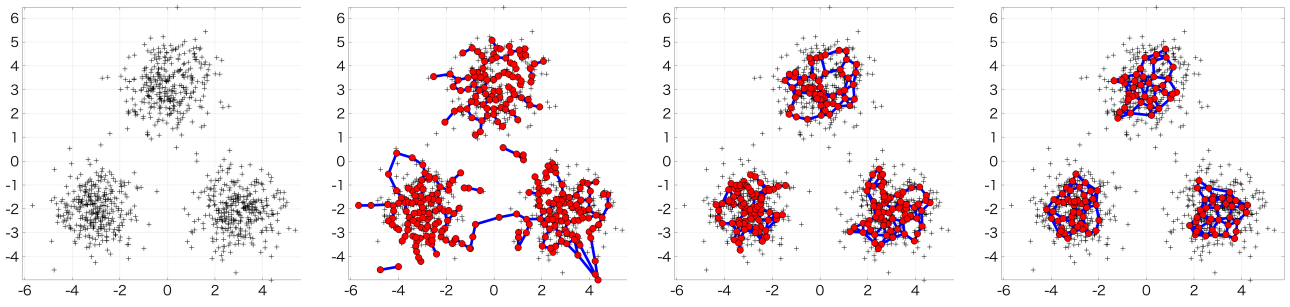
$$\begin{aligned} D_{KL}(f, g) = \quad & E_f\left[\log\left(\textstyle\sum_l^L G_{\sigma_v}(\boldsymbol{V} - \boldsymbol{v(l)})\right)\right] \\ & - E_f\left[\log\left(\textstyle\sum_j^{N_{max}} G_{\sigma_v}(\boldsymbol{W} - \boldsymbol{w_j})\right)\right], \end{aligned} \tag{12}$$

where, σ_v is the kernel bandwidth for input data, and σ_w is the one for reference vectors. Here, it can be regarded that the minimizing the second term in Eq. (12) is equivalent to optimizing the kernel bandwidth σ_w. Therefore, the cost function of kernel bandwidth σ_w is represented as follows;

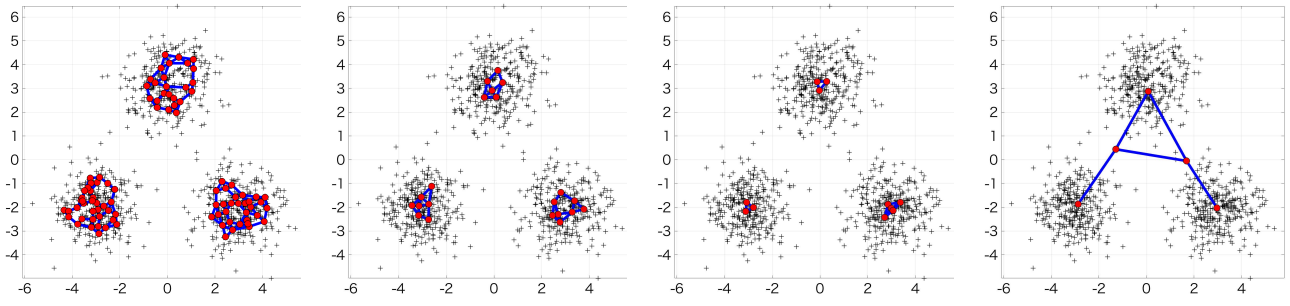
$$J(\sigma) = -\textstyle\sum_l^L \left[\log\left(\sum_j^{N_{max}} G_{\sigma_v}(\boldsymbol{W} - \boldsymbol{w_j})\right)\right]. \tag{13}$$

Here, the estimation of PDF with reference vectors can be done using a single kernel bandwidth, which is called the homescedastic case. The kernel bandwidth σ_w is calculated by gradient descent over the cost function $J(\sigma)$ as follows;

$$\frac{\partial J}{\partial \sigma_w} = -E_v\left[\frac{\sum_j^{N_{max}} G_{\sigma_w}(\boldsymbol{V} - w(j))\left[\frac{\|\boldsymbol{V} - w(j)\|^2}{\sigma_w^3} - \frac{d}{\sigma_w}\right]}{\sum_{j=1}^{N_{max}} G_{\sigma_w}(\boldsymbol{V} - w(j))}\right]. \tag{14}$$



(a) Input Data (b) Original GNG (c) GNG-CIM ($\sigma = 1.0$) (d) GNG-CIM ($\sigma = 2.0$)



(e) GNG-CIM ($\sigma = 4.0$) (f) GNG-CIM ($\sigma = 6.0$) (g) GNG-CIM ($\sigma = 8.0$) (h) GNG-CIM ($\sigma = 10.0$)

Fig. 1: Affect of the kernel bandwidth σ for self-organizing capability

TABLE I: Common parameter settings for all models

number of maximum epochs E	:	200
available maximum number of nodes N_{max}	:	300
maximum node age age_{max}	:	50
new node inserting cycle λ	:	50
error reduction rate for node inserting process α	:	0.5
error reduction rate for all nodes β	:	0.9

TABLE II: Parameter settings of Original GNG

learning rate for BMU ϵ_b	:	0.2
learning rate for connected nodes ϵ_n	:	0.05

In consideration of an on-line update rule, the stochastic gradient of Eq. (14) is applied by replacing the expected value by the instantaneous value of the input vector. Finally, the update rule of homoscedastic case is obtained as follows;

$$\sigma_w(l+1) = \sigma_w(l) - \eta_\sigma \Delta\sigma_w(l), \tag{15}$$

where, η_σ $(0 < \eta_\sigma \leq 1)$ denotes learning rate. $\Delta\sigma_w(l)$ is defined as follows;

$$\Delta\sigma_w(l) = -\frac{\sum_{j=1}^{N_{max}} G_{\sigma_w}\left(v(l)-w(j)\right)\left[\frac{\|v(l)-w(j)\|^2}{\sigma_w(l)^3} - \frac{\rho d}{\sigma_w(l)}\right]}{\sum_{j=1}^{N_{max}} G_{\sigma_w}\left(v(l)-w(j)\right)}, \tag{16}$$

where d denotes dimensions of input vector, ρ denotes a scaling factor for the stability of topological network [17].

TABLE III: Parameter settings of GNG-CIM with number of nodes based adaptation

initial kernel bandwidth σ	:	2.0
learning rate for reference vector η_w	:	0.9
CIM criterion C_{cim}	:	0.005
decreasing rate θ	:	0.1

TABLE IV: Parameter settings of GNG-CIM with node distribution based adaptation

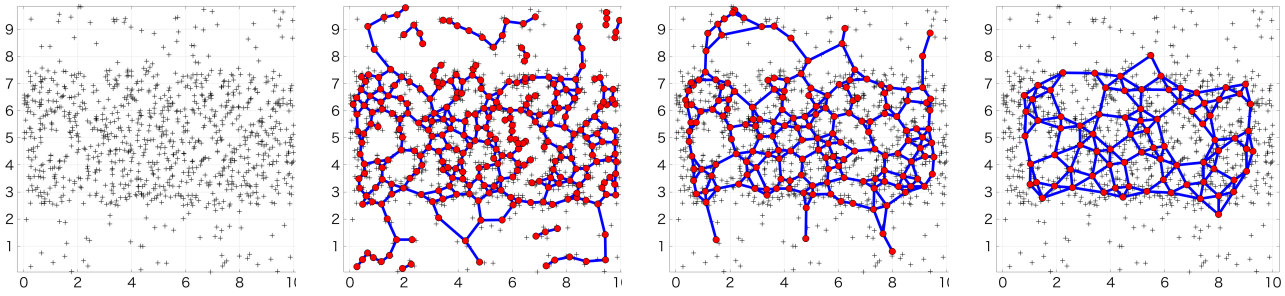
initial kernel bandwidth σ	:	10
learning rate for reference vector η_w	:	0.9
CIM criterion C_{cim}	:	0.005
learning rate for kernel bandwidth η_σ	:	0.9
scaling factor ρ	:	0.4

IV. Experiment

This section presents the simulation experiments in terms of the affect of kernel bandwidth σ in CIM, the self-organizing ability, and the quantitative comparison with Peak Signal-to-Noise Ratio (PSNR) [22] by comparing with original GNG and GNG-CIM with two kernel adaptation methods, respectively. Throughout this section, the following parameter settings are utilized; Table I shows the common parameter settings for GNG architecture. Tables II, III and IV show the specific parameter settings for each model.

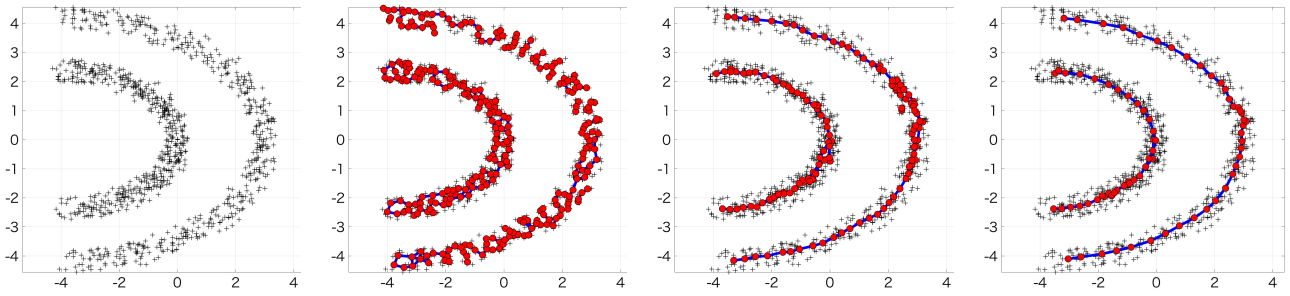
A. Affect of Kernel Bandwidth σ in CIM

First of all, we evaluate the affect of kernel bandwidth σ in CIM. Fig. 1(a) shows the input data which is consisted by



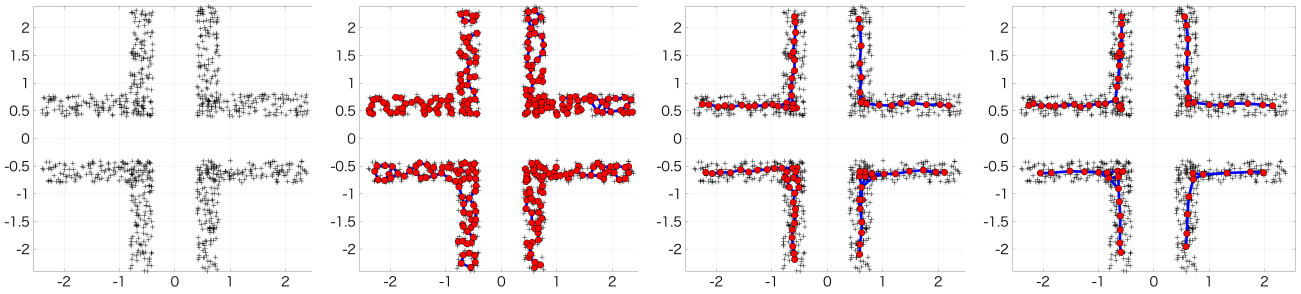
(a) Input Data (b) Original GNG (c) GNG-CIM (Num of Nodes) (d) GNG-CIM (Node Distribution)

Fig. 2: Self-organizing capability for Rectangle Data



(a) Input Data (b) Original GNG (c) GNG-CIM (Num of Nodes) (d) GNG-CIM (Node Distribution)

Fig. 3: Self-organizing capability for Half Kernel Data



(a) Input Data (b) Original GNG (c) GNG-CIM (Num of Nodes) (d) GNG-CIM (Node Distribution)

Fig. 4: Self-organizing capability for Corners Data

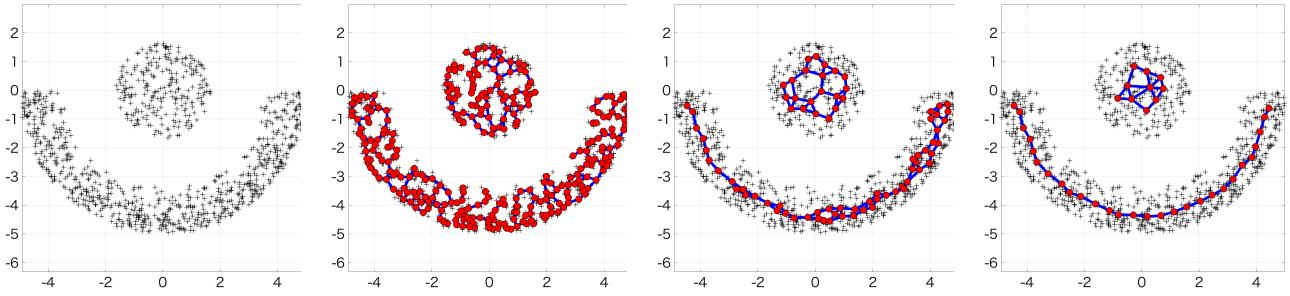
990 samples from normal distributions. For this section, the kernel bandwidth σ in GNG-CIM is fixed as Fig. 1.

In Fig. 1(b), due to an algorithm of GNG, the nodes are inserted until N_{max} and several sub clusters are organized. In contrast, GNG-CIM shows the strong outliers rejection ability. As a result, GNG-CIM is able to organize the proper number of clusters with lesser number of nodes based on the kernel bandwidth. In other words, we could be designed the information based on the appropriate dimension reduction. Furthermore, the results in Fig. 1 emphasize that it is significant to define the appropriate kernel bandwidth to obtain useful information for further processing.

B. Self-Organizing Experiment

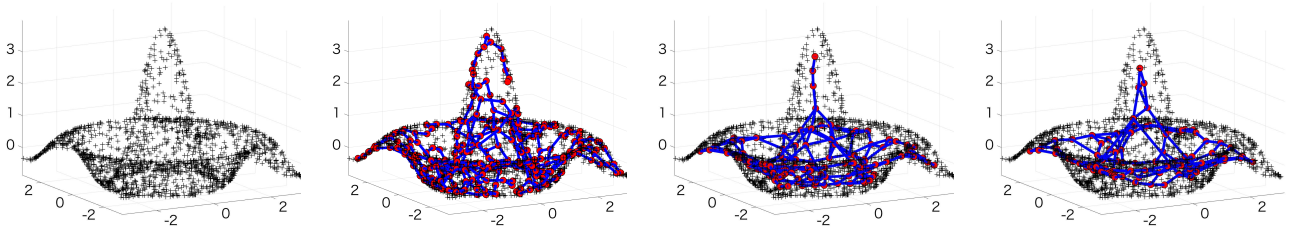
This experiment evaluates the self-organizing capability of models with respect to 2D and 3D data distributions. In this experiment, each dataset is consisted by L=1000 data points, respectively. Figs. 2, 3, 4 and 5 show the results of 2D dataset of Rectangle, Half Kernel, Corners and Crescent Moon Data, respectively. Fig. 6 shows the result of 3D Wave data. Here, only the Rectangle Data (Fig. 2) is including the noise information.

In Fig. 2, original GNG is greatly influenced from noise information. On the other hand, GNG-CIM (especially, node



(a) Input Data (b) Original GNG (c) GNG-CIM (Num of Nodes) (d) GNG-CIM (Node Distribution)

Fig. 5: Self-organizing capability for Crescent Moon Data



(a) Input Data (b) Original GNG (c) GNG-CIM (Num of Nodes) (d) GNG-CIM (Node Distribution)

Fig. 6: Self-organizing capability for 3D Wave Data

distribution based adaptation method) shows the superior outlier rejection ability. This is because the CIM is a probability distributions based function. The influence from node in low probability (density) region is lower than the one from high probability region. Due to the algorithm of original GNG, the nodes are inserted as much as possible until N_{max}. Thus, there are lots of sub clusters, especially Fig. 3. In contrast, GNG-CIM have clustered data into appropriate number of clusters without having sub clusters. In regards to 3D dataset as Fig. 6, the nodes in original GNG cover the surface of data. However, the network structure is not well represented the shape of data (Several edges are not connected properly). On the other hand, the network of GNG-CIM represents the summary of data properly although the nodes in network are not covered the surface of data.

From the self-organizing experiment, it can be regarded that CIM-GNG has the superior self-organizing capability with smaller number of nodes in topological network. In addition, there is a possibility to improve the ability of GNG-CIM by finding the more suitable kernel bandwidth for data.

C. Quantitative Comparison Experiment

In order to show a quantitative comparison for the unfolding capability, the PSNR of topological network is compared. The PSNR is calculated as follows [22];

$$\text{PSNR} = 10\log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right), \tag{17}$$

where, MAX_I^2 is the squared euclidean norm of input vector $\boldsymbol{V}$, MSE denotes mean squared error which is given as follows;

$$\text{MSE} = \frac{1}{L}\sum_{i=1}^{L}\|w(i) - v(i)\|^2, \tag{18}$$

here, L denotes the number of input vectors, $v(i)$ is i-th input vector, and $w(i)$ represents the winning reverence vector corresponding to $v(i)$.

According to the definition of PSNR, it can be regarded that the model with higher PSNR is better. In addition, we also consider the number of nodes N in topological network for the evaluation of unfolding capability. Thus, it can be regarded that the model with higher PSNR and smaller number of nodes has a superior ability.

The results of quantitative comparison are shown in Fig. 7. In this experiment, we utilized the results from previous experiment to calculate PSNR and number of nodes N in topological network. In Fig. 7, the number of nodes in original GNG shows higher states than GNG-CIM models although it shows better PSNR. Here, the number of nodes in original GNG is inserted until N_{max}, thus Log(N) for each dataset shows same value. CIM based models show better dimensional

reduction performance than other models. Thus, the models have smaller Log(N). Comparing with PSNR, GNG-CIM with number of nodes based adaptation shows lower state than original GNG. On the other hand, GNG-CIM with node distribution based adaptation shows similar or higher PSNR, except for Corners dataset. This is because the number of nodes in network is too low. According to Eq. (17), the large number of nodes makes high PSNR. Thus, there is a possibility to improve the performance by changing the parameter settings relating to CIM.

The results from this section, it can be considered that GNG-CIM with node distribution based adaptation has the superior abilities than other models.

V. Conclusion

In this paper, we introduced the generalized similarity measurement called CIM to GNG instead of Euclidean-based one in GNG, and we utilized CIM for the criteria of node insert process. CIM is a kernel method based similarity measurement, thus we also introduced the kernel bandwidth adaptation algorithms based on the approaches in SOM-CIM. The experimental results show that due to the adaptation of the kernel bandwidth, GNG-CIM is able to control the complexity of model easily. Self-organizing capability in GNG-CIM shows that the model is able to represent the data structure with the less number of neurons than original GNG, with maintaining the outstanding outliers rejection ability.

As a future work, we will consider the optimization algorithm for kernel bandwidth adaptation and the number of nodes, for maximizing the performance of GNG-CIM.

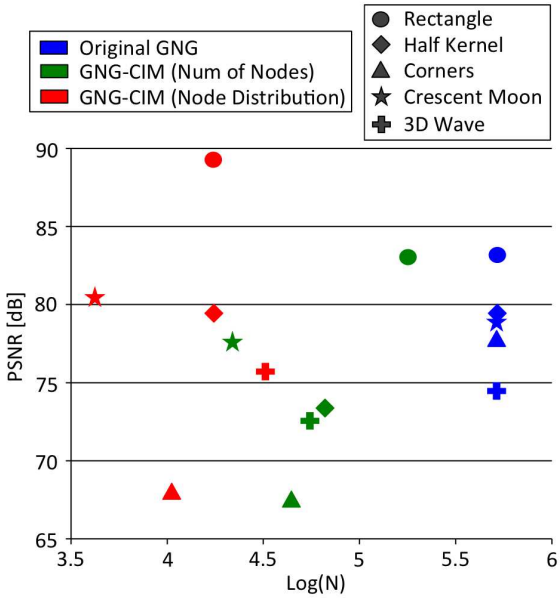


Fig. 7: Results of quantitative comparison based on PSNR with 4 types of data set. Data points near top left-hand corner (higher PSNR with lower number of nodes N) stand for the better unfolding capability.

Acknowledgment

The authors would like to acknowledge a FRGS grant (Project No.: FP069-2015A) from Ministry of Higher Education in Malaysia.

References

[1] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[2] D. Tibaduiza, L. Mujica, and J. Rodellar, "Damage classification in structural health monitoring using principal component analysis and self-organizing maps," *Structural Control and Health Monitoring*, vol. 20, no. 10, pp. 1303–1316, 2013.

[3] P. H. Niknam, B. Mokhtarani, and H. Mortaheb, "Prediction of shock-wave location in supersonic nozzle separation using self-organizing map classification and artificial neural network modeling," *Journal of Natural Gas Science and Engineering*, vol. 34, pp. 917–924, 2016.

[4] N. Chen, B. Ribeiro, A. Vieira, and A. Chen, "Clustering and visualization of bankruptcy trajectory using self-organizing map," *Expert Systems with Applications*, vol. 40, no. 1, pp. 385–393, 2013.

[5] A. Majumder, L. Behera, and V. K. Subramanian, "Emotion recognition from geometric facial features using self-organizing map," *Pattern Recognition*, vol. 47, no. 3, pp. 1282–1293, 2014.

[6] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52–65, 2013.

[7] B. Fritzke, "A growing neural gas network learns topologies," *Advances in neural information processing systems*, vol. 7, pp. 625–632, 1995.

[8] D. Viejo, J. Garcia-Rodriguez, and M. Cazorla, "Combining visual features and growing neural gas networks for robotic 3d slam," *Information Sciences*, vol. 276, pp. 174–185, 2014.

[9] S. Orts-Escolano, J. Garcia-Rodriguez, V. Morell, M. Cazorla, J. A. S. Perez, and A. Garcia-Garcia, "3d surface reconstruction of noisy point clouds using growing neural gas: 3d object/scene reconstruction," *Neural Processing Letters*, vol. 43, no. 2, pp. 401–423, 2016.

[10] C. A. T. Mendes, M. Gattass, and H. Lopes, "Fgng: A fast multi-dimensional growing neural gas implementation," *Neurocomputing*, vol. 128, pp. 328–340, 2014.

[11] T. Vierjahn and K. Hinrichs, "Surface-reconstructing growing neural gas: A method for online construction of textured triangle meshes," *Computers & Graphics*, vol. 51, pp. 190–201, 2015.

[12] M. Ghesmoune, M. Lebbah, and H. Azzag, "A new growing neural gas for clustering data streams," *Neural Networks*, vol. 78, pp. 36–50, 2016.

[13] A. Boonmee, K. Sethanan, B. Arnonkijpanich, and S. Theerakulpisut, "Minimizing the total cost of hen allocation to poultry farms using hybrid growing neural gas approach," *Computers and Electronics in Agriculture*, vol. 110, pp. 27–35, 2015.

[14] E. Mohebi and A. Bagirov, "Modified self-organising maps with a new topology and initialisation algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 27, no. 3, pp. 351–372, 2015.

[15] E. J. Palomo and E. López-Rubio, "Learning topologies with the growing neural forest," *International journal of neural systems*, vol. 26, no. 04, p. 1650019, 2016.

[16] S. Y. Kung, *Kernel methods and machine learning*. Cambridge University Press, 2014.

[17] R. Chalasani and J. C. Principe, "Self-organizing maps with information theoretic learning," *Neurocomputing*, vol. 147, pp. 3–14, 2015.

[18] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: properties and applications in non-gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[19] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[20] D. Erdogmus, R. Jenssen, Y. N. Rao, and J. C. Principe, "Gaussianization: An efficient multivariate density estimation technique for statistical signal processing," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 45, no. 1-2, pp. 67–83, 2006.

[21] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[22] D. Salomon, *Data compression: the complete reference*. Springer Science & Business Media, 2004.