

A constrained growing grid neural clustering model

Chihli Hung

Published online: 10 January 2015
© Springer Science+Business Media New York 2015

Abstract A novel extension of the growing grid (GG) algorithm is proposed in this paper. The learning behavior of the traditional GG model is affected by two factors i.e. similarity between output units and their associated input vectors, and lateral connections between output units. Based on the assumption that the more active unit should move more towards its associated input vector, the constrained GG emphasizes the effect of the lateral connections between output units in a grid, and neutralizes the effect on the distance between the input vector and neighbors of the best matching unit (BMU). A constrained learning rule is tested by fifteen data sets, i.e. the square, animal, iris, ionosphere, sentiment polarity data sets and the fundamental clustering problem suite (FCPS), which includes ten data sets. Based on five evaluation measures, i.e. average quantization error, error entropy, BMU activation rate, average map scope and topographic error, the performance of GG is improved if the constrained learning rule is used. In addition, we use the *t*-test to test whether or not the proposed models outperform the traditional GG significantly. Except in some cases, the experiments conclude that the proposed approach can significantly improve the traditional GG model based on five evaluation criteria for fifteen data sets.

Keywords Self-organizing maps · Growing grid · Constrained learning rule · Neural clustering · Competitive neural learning

1 Introduction

This paper proposes a novel growing grid approach, namely the constrained growing grid (CGG) for organization of high-dimensional data by using the self-organizing map-like technique with a constrained learning function. The self-organizing map (SOM), which is proposed by Teuvo Kohonen [1], is a powerful clustering algorithm for the visualization of high-dimensional data. It projects the high-dimensional data onto a low-dimensional map, usually a two-dimensional grid of units. The geometric relationship between units in a grid represents the relationship between high-dimensional data. In other words, the SOM is able to abstract the most important data relationships in order for them to be visualized on a two-dimensional map, which is a robust tool for various tasks such as data organizing, machine monitoring, pattern recognition, data mining, image analysis, communication, market segmentation, cloud resources management, 3-D shape reconstruction etc. e.g., [2–9].

The SOM depends significantly on certain predefined parameters including a fixed topological structure, which may be difficult to decide before learning. This has inspired the development of many extended SOM-like models such as growing cell structure (GCS) [10], growing neural gas (GNG) [11], growing neural gas with utility (GNG-U) [12], grow when required (GWR) [13], dynamic adaptive self-organizing hybrid (DASH) [14], self-organizing incremental neural network (SOINN) [15], local adaptive receptive field self-organizing map (LARFSOM) [16], and growing self-reconstruction map (GSRM) [6]. A common goal of these extended models is to project a data set from a high-dimensional space onto a low-dimensional space, and keep its inner structure as faithful as possible. Although these extended SOM-like models are successful in conquering the

C. Hung (✉)
Department of Information Management, Chung Yuan
Christian University, Zhongli, Taiwan
e-mail: chihli@cycu.edu.tw

shortcoming of predefinition for the fixed topological structure, their non-grid topological feature makes them difficult to present on a two-dimensional space without information loss and thus losing the benefits of visualization.

Unlike such non-grid extended SOM-like models, Fritzke [17] proposed growing grid (GG), which is an incremental variant of the SOM in terms of the network topology. GG develops a growing grid structure based on an input data set without using the predefined topological structure. Thus, GG modifies the static SOM into a dynamic SOM and keeps a two-dimensional grid as its topological structure for visualization. The best matching unit (BMU) of GG is generally the most active unit and has the greatest reaction to its associated input vectors from the theoretical viewpoint of self-organization [18]. However, the learning function of the traditional GG may not make the BMU the most active unit during learning, as this goes against the competitive learning principle [19]. This paper modifies the original learning function of the GG and proposes the constrained growing grid (CGG) to pursue a better performance than the traditional GG model.

The remainder of this paper is organized as follows. In Section 2, we briefly review related literature, including neural gas (NG), growing cell structure (GCS), growing neural gas with utility criterion (GNG-U), grow when required (GWR), dynamic adaptive self-organizing hybrid (DASH), growing self-reconstruction maps (GSRM) and other recent related work. Sections 3 and 4 introduce the growing grid growing process and learning process respectively. In Section 5, we propose the constrained neural learning algorithm. Section 6 introduces five evaluation measures used in this paper, which are average quantization error (AQE), error entropy, average map scope (AMS), BMU activation rate (BAR) and topographic error (TE). Section 7 shows the experiment design and results. Finally, the conclusion and possible future work are provided in Sections 8 and 9 respectively.

2 Related work

Due to the deficiency of the SOM, i.e. the need for a fixed topological structure before learning, several extended SOM-like neural learning models have been proposed [6, 10–14, 16]. Different modifications of the SOM suggest different enhancements from different viewpoints. For example, Martinetz and Schulten [20] proposed the neural gas (NG) model to alleviate the requirement of prior knowledge about the topological structure of the SOM. Like the SOM, the NG model defines a fixed number of output units and a learning rate which decays over time during learning. Unlike the SOM, the NG model defines the relationships between output units by connecting the best matching unit

(BMU) and the second best matching unit (SMU) to an input vector while learning, which is called competitive Hebbian learning (CHL) [21]. All output units are updated based on their relative order, which is measured by the distance compared with other output units, rather than the absolute distance, to the input vector. Without the topological constraint, the NG units act like a gas, and spread throughout the input space.

The growing cell structure (GCS) is a dynamic neural model that maintains its units with a triangular connectivity [10]. GCS starts with three units and increases by one unit after every predefined period. A new unit is inserted by splitting the farthest unit from the unit with the greatest error. A unit with a low probability density, meaning that few input vectors are mapped onto it, will be removed, together with its direct neighbors on the corresponding triangle. The growing neural gas (GNG) [11] is a neural model applying the GCS growing mechanism with the CHL topology. GNG starts with two units and connects an input vector's BMU to SMU. At each learning iteration, the age variable of each connection that is directly linked to the BMU is increased by 1, but the age variable of the connection between BMU and SMU is initialized to zero. After every predefined period, a new unit is inserted by splitting the unit with the highest error in the direct neighborhood from the unit with the highest error in the whole structure. The unit-pruning function works through the connection-trimming function, which removes old connections that exceed a predefined threshold and units which have no neighbors. Both GCS and GNG have two fixed learning rates which are applied to BMU and its direct neighbors respectively.

Fritzke [12] proposed the growing neural gas with utility criterion (GNG-U), which is an extension of GNG with an ability to handle rapid changes of input patterns. The GNG-U contains all GNG features but employs an extra unit-pruning function. GNG-U uses the utility of a unit to evaluate the contribution of this unit to the topological structure. A utility is an accumulated value, which computes the difference between the distance to BMU and SMU for an input vector. In other words, the utility of a unit is the amount by which the global error is increased if this unit is removed. Where a unit has small utility this means that it is not activated frequently, or that it can be replaced by the SMU with no harm to the model. If the proportion of biggest error to smallest utility in a model is greater than the utility-removing threshold, this unit will be pruned.

Marsland et al. [13] argued that a model should grow because it cannot represent the input vectors well rather than simply because a fixed predefined period is achieved. They propose the grow when required (GWR) model to discover the variation in a data set. GWR starts with two units and applies a CHL topology. At each iteration, all connections to the BMU are aged by 1, but the age for the connection

between BMU and SMU is initialized to zero. A unit is inserted near to the BMU because the distance to the BMU for an input vector is greater than the unit-growing threshold. GWR also offers a unit-pruning function, which is performed through the removal of aged connections.

As selecting the appropriate thresholds for modification of the topological structure is a difficult task, Hung and Wermter [14] proposed the dynamic adaptive self-organizing hybrid (DASH), which contains a dynamic structure, hierarchical training, non-stationary data learning and parameter self-adaption. DASH starts with two units and stops when all units represent their associated input vectors well, or the number of inputs is too few to build a sub-map. The recursive learning continues for the individual unit whose average quantization error does not meet the requirement of DASH. DASH uses a CHL topology to connect BMU and SMU for an input stimulus. A connection is trimmed if it is relatively old compared to other connections and a unit without any connection is removed. However, if the model has met its quality requirement, the connection-trimming function is restrained.

Do Rêgo et al. [6] proposed the growing self-reconstruction map (GSRM), which is extended from the GNG model for use in the field of three-dimensional shape reconstruction. GSRM starts with three units randomly chosen from the input vectors. The main difference between GSRM and GNG is that GSRM uses an extension of the CHL to update the connections between units. The original CHL simply builds the connection between BMU and SMU, but the extended competitive Hebbian learning (ECHL) builds and removes the connection in order to produce a triangular two-manifold mesh representation of a three-dimensional target object. More complete literature reviews in the field of SOM with a time-varying structure can be found in [22].

Although many extended SOM-like models have successfully used a time-varying structure in order to relieve the need for a predefined topological structure before the presentation of input data, their non-grid topological feature prevents them from being presented on a two-dimensional space without information loss. Therefore, these non-grid SOM-like models lose the visualization benefit of the

original SOM, which is its major feature compared with other clustering techniques.

Unlike such non-grid SOM-like models, growing grid (GG) is an incremental variant of the SOM in terms of the network topology [17]. By keeping a grid structure, GG uses a competitive-based learning technique, which aims to make the BMU learn more than its neighbors for each input vector and represent the relationship between input vectors on a two-dimensional map without information loss. However, the extent of modification is decided not only by the geometric relationships between the BMU and its neighbors but also by the distance between the input vector and its associated output units. Consequently, the BMU may not always move more towards its associated input vector because the distance between the BMU and this input vector is the shortest. Therefore, we propose a constrained learning function for GG to obey the competitive learning principle [19, 22, 23].

3 The GG growing process

Like the SOM, GG is a neural clustering approach. More specifically, GG is a dynamic variant of the SOM. GG contains growing and fine-tuning stages, which use the same learning process but different learning rates: a fixed learning rate at the growing stage and a decaying learning rate over time at the fine-tuning stage. The growing process of GG is achieved by increasing rows of units or columns of units in a grid. It starts with 2x2 units in a grid architecture and develops the grid periodically by inserting a whole set of units in a column or row (Fig. 1). The units grow from the position between the most frequently activated unit and its farthest direct neighboring unit (Fig. 1b). Thus, by keeping a grid structure that is formed by the positional units, GG inherits from the SOM a method of providing a two dimensional view effortlessly. The growing frequency depends on the learning length for the target unit number as shown in (1). After g_t number learning steps, the whole set of units in a column or row is inserted. Criteria for stopping growth should be set in order to prevent the grid from growing indefinitely. Like the SOM, the GG uses the

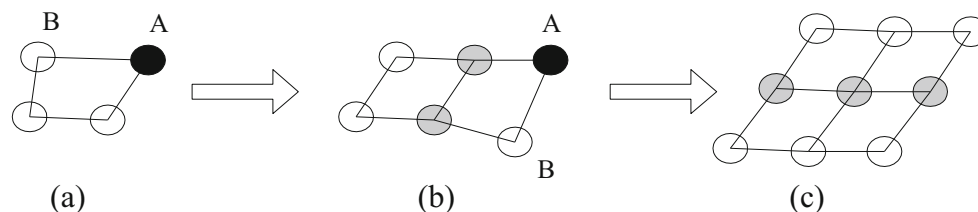


Fig. 1 The growing process of the Growing Grid (GG) model. GG adds a set of units after every g_t learning steps as in (1). Units are represented as circles. Lateral connections between units are represented

as lines. Circle A denotes the most frequently activated unit and Circle B is the farthest direct neighbor for Circle A. Grey circles are new units at each growing iteration

predefined learning length as its natural stop criteria. Unlike the SOM, which uses two predefined learning lengths for rough learning and fine-tuning learning respectively, the GG starts its fine-tuning learning stage once its unit number achieves a predefined target number of units at the growing stage [17].

$$g_t = \max \left(1, \text{ceil} \left(\frac{l}{U - u_t} \right) \right), \tag{1}$$

where l is the learning length, U is the target unit number, t is time, and u_t denotes the number of units built at time t .

4 The GG learning process

The learning process is the means by which units of the grid are modified to represent their associated input samples as faithfully as possible. Like the SOM, GG uses a topological structure of units such that adjacent units contain similar weights so that they self-organize into an ordered map [1]. Based on a current topological structure, units in the output layer are interconnected with lateral connections. This interconnection influences the difference in activation for output units during learning.

At each learning step, the first core rule is to decide the best matching unit (BMU) for a current input vector. By comparison with an input vector, units of current GG topology compete against each other. The winner is the BMU, which is the unit most similar to the input vector and which represents the input vector best, as in (2). The second core rule is to make this BMU together with its neighbors, move towards the input vector. The BMU should then be the unit with the largest activation value, and units in a neighborhood with close proximity are more active than those units in a neighborhood that is further away.

$$\|X(t) - W_b(t)\| \leq_{j=1}^K \|X(t) - W_j(t)\| \tag{2}$$

where $\| \cdot \|$ is the Euclidean norm, X is the input vector, W is the output unit vector, t denotes time, K is the total number of output units in the current topological grid and b is the BMU.

In Fig. 2, the input vector is represented as circle X . Units in neighbourhood 1 are more active than those in neighbourhood 2. The learning process is a way of making these units represent input vectors better. In other words, as in the concept of vector quantization, the learning process is a way to reduce the quantization error. This is carried out by a neighborhood function and learning rate in the update function of the GG model, as in (3). At the growing stage, the GG

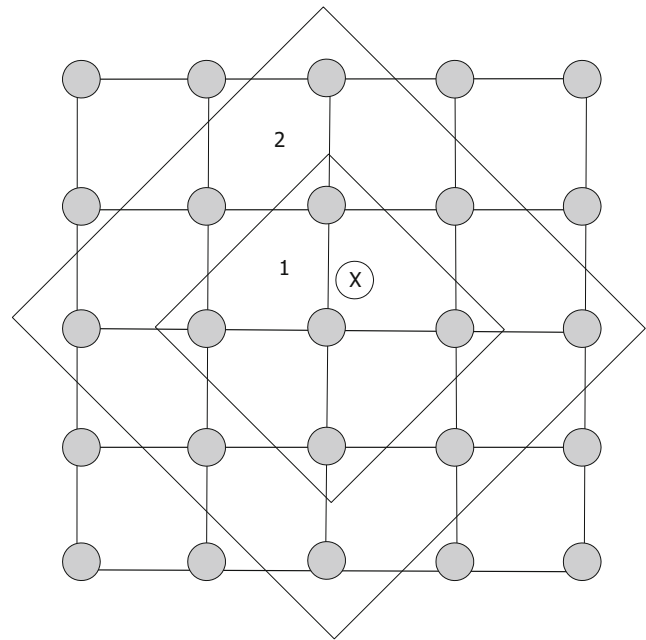


Fig. 2 An example of the neighborhood. The input vector is represented as circle X . Units in neighborhood 1 are more active than those in neighborhood 2

model uses a fixed learning rate to represent its associated input samples quickly. At the fine-tuning stage, the topological structure is fixed. The GG model fine-tunes its output

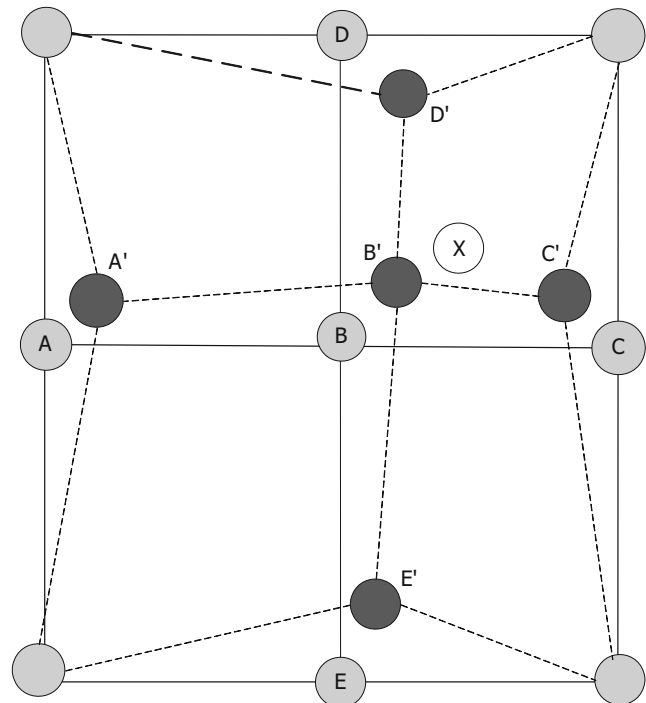


Fig. 3 An example of the 3x3 GG learning influenced by a current topological structure. The winner B and it neighbors A, C, D and E move towards the input vector X and the modified units are shown as dark circles

units with a time decaying learning rate in order to pursue a better representation.

$$W_j(t + 1) = W_j(t) + \alpha(t)A_b(j, t)(X(t) - W_j(t)), \quad (3)$$

where t denotes time, $\alpha(t)$ is the learning rate function and $A_b(j, t)$ is a bell-shaped neighborhood function centered on the BMU b as (4), X is the input vector, and W is the output unit vector.

$$A_b(j, t) = e^{-d_{bj}^2/2\delta_t^2}, \quad (4)$$

where δ_t is the neighborhood radius at time t and $d = |r_b - r_j|$ is the city-block distance between output units b and j on the grid.

In Fig. 3, before learning, output units are denoted as light grey units, such as units A, B, C, D, E , etc. Unit B is the best matching unit (BMU) because it is the nearest unit to the input vector X . In this example, only the winner and its direct neighbors are updated. That is, the winner B and its neighbors A, C, D and E move towards the input vector X .

5 Constrained neural learning

The GG algorithm is a robust tool for a mapping task but it has also been proposed as a biological model of topological map formation. From the viewpoint of biological function, the competition feature enhances the winning unit’s reaction, while the cooperation feature makes the BMU and its neighbors work together towards modification of the synaptic strengths [24]. The lateral connection of output units is crucial for self-organization and formation of a topologically ordered network. The topologically ordered network is defined and proven by [25, 26] as in (5).

$$A_b(b1, t) > A_b(b2, t) \text{ if } \|X(t) - W_{b1}(t)\| < \|X(t) - W_{b2}(t)\| \quad (5)$$

where $A_b(b, t)$ is the neighborhood function centered on the unit b and $\| \cdot \|$ is the Euclidean norm.

According to the GG algorithm, the update rule (3) forces each output unit within the defined neighborhood to move towards the input vector, by the distance between the output unit and the input vector. However, besides the neighborhood function, this distance also represents the strength of activation. Compared to its neighbors, the BMU may not move the most towards the input vector since the BMU is the closest unit to the input vector. In other words, the BMU may not always be the most responsive unit to a given input vector. Although the neighborhood function reduces this opportunity, it may make output units overactive in the BMU direct neighborhood.

On the other hand, from a biological viewpoint, neurons or output units do not change their behavior abruptly but in a smooth way [27]. The GG adapted methods will make output units go back and forth at the learning stage so that the cortical column may not spread suitably. Thus, we constrain the output unit adapted behavior to the area of the input vector and its associated BMU. The modification of the traditional GG algorithm is proposed as in (6). As unit b is the best matching unit, its Euclidean norm or distance to the input vector X at time t (i.e., $\|X(t) - W_b(t)\|$) should be the shortest among the units. In (6), the activation of neighbors of the BMU (i.e. unit j , where $j \in b$'s neighbors, $j \neq b$) is constrained to the proportion of $\|X(t) - W_b(t)\|$ to $\|X(t) - W_j(t)\|$. If $X(t) \neq W_j(t)$ and $j \neq b$, then $\frac{\|X(t) - W_b(t)\|}{\|X(t) - W_j(t)\|}$ is less than 1. If $X(t) \neq W_j(t)$ and $j = b$, then $\frac{\|X(t) - W_b(t)\|}{\|X(t) - W_j(t)\|}$ equals 1. Therefore, this modification (6) will guarantee that the BMU will have the strongest activation to the input vector and there is no influence for the BMU *per se*.

$$W_j(t + 1) = \begin{cases} W_j(t) + \alpha(t)A_b(j, t)(X(t) - W_j(t)) \frac{\|X(t) - W_b(t)\|}{\|X(t) - W_j(t)\|}, & \text{if } X(t) \neq W_j(t), \\ W_j(t), & \text{otherwise} \end{cases} \quad (6)$$

where t denotes time, $\alpha(t)$ is the learning rate function, $A_b(j, t)$ is the neighborhood function centered on the BMU b .

6 Evaluation measures

In this research, the average quantization error (AQE) [3], error entropy [28], average map scope (AMS) [23], BMU activation rate (BAR) and topographic error (TE) [3, 29] are evaluated for the explanation ability, smooth hyper surface, border effect, BMU activation and topology preservation of the GG algorithms respectively.

6.1 Average quantization error

The quantization error, which is also called the distortion measure for SOM-like models, is suggested by Kohonen as a measurement used in the vector quantization technique [3]. The quantization error is defined as the sum of the Euclidean norm between every input vector and its BMU. A trained topological structure with a smaller value of quantization error produces a smaller distortion for their input samples. Thus, a better GG model should have a smaller quantization error. As the number of output units affects the quantization error, this paper uses the average quantization

error (AQE) instead. Given an input vector X_i , the average quantization error is described as in (7).

$$AQE = \frac{1}{N} \sum_{i=1}^N \|X_i - W_i\|, \tag{7}$$

where N is the total number of input samples, X_i is the weight vector of input sample i , and W_i denotes the BMU of X_i .

6.2 Error entropy

The GG models use their trained grids to represent input vectors. An ideal SOM-like model should build a smooth hyper surface and each unit on the surface should be evenly distributed [3]. Based on this concept, Wu and Takatsuka [28] proposed a measure, namely error entropy, to evaluate the average distribution of error for units on the map. Entropy [30], from information theory, is a measure to evaluate the uncertainty of variables, which is able to evaluate the degree of uniformity of the error distribution on the map. This paper uses the same definition of error entropy as in the work of Wu and Takatsuka [28] and Hung [23], which is shown in (8). A map with greater error entropy has a more even error distribution, producing a smoother surface to represent input vectors. A unit that does not represent any input samples is given an error entropy of zero.

$$ErrorEntropy = - \left(\sum_{j=1}^K e_j \log_2 e_j \right)$$

where
$$\begin{cases} e_j = \frac{UQE_j}{TQE}, & \text{if } UQE_j > 0 \\ e_j = 1, & \text{if } UQE_j = 0 \end{cases}, \tag{8}$$

K is the total number of units, UQE_j and TQE mean the quantization error of unit j and total quantization

error for the map, which are shown in (9) and (10), respectively.

$$UQE_j = \sum_{i=1}^L \|X_i - W_j\|, \tag{9}$$

where W_j is the weight vector of unit j , which is the BMU for input sample i , and L is the total number of input samples whose BMU is unit j .

$$TQE = \sum_{i=1}^N \|X_i - W_i\|, \tag{10}$$

where X_i is the weight vector of input sample i , W_i is the weight vector of BMU for input sample i and N is the total number of input samples.

6.3 Average map scope

One well-known problem of the SOM-like model is a border effect, whereby units at the border of a trained map do not represent their associated input samples well compared with those units inside the map [3, 23]. The border effect happens because units at the border of the map have fewer neighbors than those inside the map. Thus, the whole map does not stretch out enough. The result of this effect is that input samples whose BMUs are at the border are not represented well compared with those samples whose BMUs are inside the trained map. For example, in Fig. 4a and b, each input sample is illustrated by a small grey dot and each output unit is represented by a circle. The trained map in Fig. 4b stretches wider than that in Fig. 4a, so the border effect in Fig. 4a is greater than that in Fig. 4b. Hung [23] proposed the average map scope (AMS) to evaluate the border effect, which is also used in this research. AMS is defined as the average distance of border units to the center of the map as

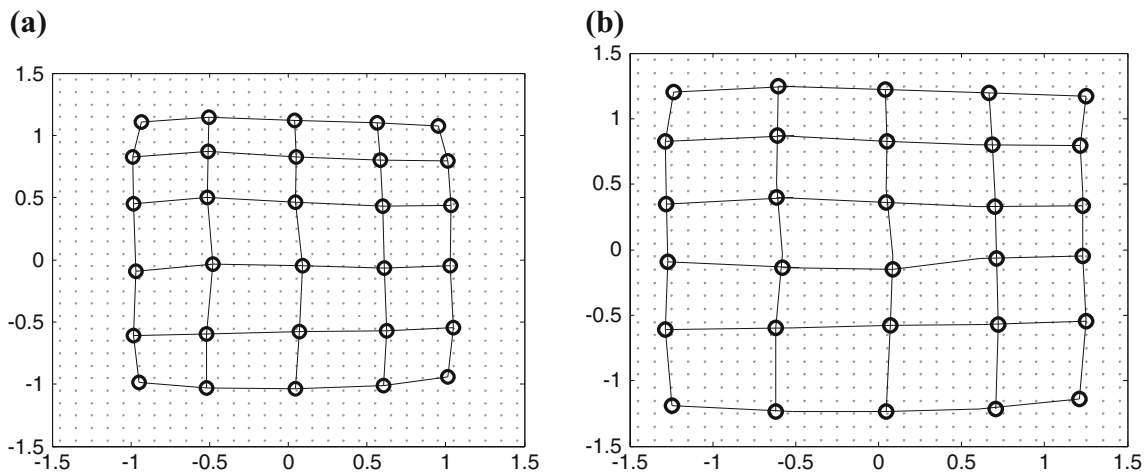


Fig. 4 (a) A 6x5 GG with a greater border effect. (b) A 6x5 GG with a smaller border effect

in (11). A trained map of GG with a smaller border effect should have a larger average map scope.

$$\text{Average map scope} = \frac{1}{B} \sum_{i=1}^B \|W_i - W_c\|, \tag{11}$$

where B is the total number of border units, W_i denotes the weight vector of the border unit, W_c denotes the average weight vector of all units, as shown in (12).

$$W_c = \frac{1}{K} \sum_{j=1}^K W_j, \tag{12}$$

where K is the total number of units and W_j is the weight vector of unit.

6.4 The BMU activation rate

As the traditional GG algorithm follows the competitive learning principle [17], the unit which is closest to the input vector should be more active than the others. However, according to the original update rule as shown in (3), the BMU may not produce the greatest response to the input stimulus due to the distance between the BMU and its associated input vector, which is the shortest. Although the bell-shaped neighborhood function as shown in (4) may reduce this situation, it is interesting to test whether the BMU moves more than its neighbors towards their associated input vector during learning. This test can be done by the BMU activation rate (BAR), which is shown in (13).

$$\text{BAR} = \frac{1}{S} \sum_{i=1}^S r(x_i),$$

$$r(x_i) = \begin{cases} 1, & \text{if the BMU moves the most towards } x_i \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where S is the total number of learning steps. The value of $r(x_i)$ equals 1, if the BMU moves the most towards its associated input vector x_i , or otherwise equals 0.

6.5 Topographic error

Like the SOM, one main goal of a GG model is to project input vectors from a high-dimensional space onto a low-dimensional space, and faithfully keep their internal structure. Therefore, nearby input vectors are projected onto output units, which are located nearby in the output space. For example, if two input vectors X and Y have a neighborhood relationship, their BMUs (i.e. W_x and W_y) should also be neighbors. This characteristic of the projection preserves the topology of data and is a useful evaluation criterion for the quality of neural clustering [31]. There are several measures to evaluate topology preservation, such as topographic product [32], topographic function [33], topographic error

[29] and some other alternatives [34]. Generally speaking, the tests for topology preservation require huge computational resources for the calculation of the distances between all units and the distances between all input vectors. This research uses topographic error to measure the topology preservation as this approach reduces the huge computational burden, as suggested by Kohonen [3]. Topographic error is defined as the proportion of all data vectors whose first and second BMUs are not adjacent units (14). The lower the topographic error is, the better the GG preserves the topology.

$$TE = \frac{1}{N} \sum_{i=1}^N u(x_i), u(x_i)$$

$$= \begin{cases} 1, & \text{if the 1}^{st} \text{ and 2}^{nd} \text{ BMUs are not neighbors} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where *neighbor* denotes units with the direct connection between each other, N is the total number of input samples and x is the input vector.

7 Experiments

To test the performance of the traditional GG and the proposed constrained growing grid (CGG), we use fifteen data sets, including a square data set [23], animal data set [35], iris data set [36], ionosphere data set [37], sentiment polarity data set [38, 39] and the fundamental clustering problem suite (FCPS) [40], which consists of ten data sets (i.e., Atom, Chain Link, Engy Time, Golf Ball, Hepta, Lsun, Target, Tetra, Wing Nut, Two Diamonds). The whole learning length includes two stages, i.e. the growing and fine-tuning stages. The GG model grows its units only at the growing stage with a fixed and greater learning rate. In the second phase, the GG model fine-tunes its current topological structure with a smaller learning rate, which decays over time. We extend the traditional GG model by two approaches to test our constrained neural learning rule. The first one, namely fine-tune CGG, keeps the same learning rule as that in the traditional GG at the growing stage and constrains the reaction of non-BMUs to their associated input vectors only at the fine-tuning stage. Thus we can evaluate the performance of the constrained neural learning rule in a stable topological structure of GG. The second approach, namely full CGG, uses the constrained neural learning rule at both growing and fine-tuning stages. The full CGG realizes a real winner-take-more neural learning model, in which the best matching unit has the strongest reaction to its associated input stimulus.

For comparison, we treat the traditional GG as the benchmark model. The traditional self-organizing map (SOM)

Table 1 The Performance Comparison Evaluated by AQE, error entropy, BAR, AMS and TE for a square data set

Models	AQE	Error Entropy	BAR	AMS	TE
SOM	0.315	4.3105	0.05	1.0341	0
CSOM	0.257	4.5489	1	1.1932	0
GG	0.279	4.467	0.182	1.106	0.011
Fine-tune CGG	0.282 (0.235)	4.436 (0.175)	0.632** (0.000)	1.116 (0.217)	0.021 (0.160)
Full CGG	0.256** (0.000)	4.511 (0.181)	1** (0.000)	1.234** (0.000)	0.008 (0.052)

The value in bold form denotes a better performance than its benchmark. The p-value of t-test is shown in parenthesis.

*denotes that the difference between two models achieves the 0.05 significance level.

** denotes that the difference between two models achieves the 0.01 significance level.

and its constrained version (i.e. CSOM) are also provided. All growing grid models use the same parameters such as the learning rate, the learning length and the target unit number. In this paper, the fixed learning rate at the growing stage is 0.1 and the initial decaying learning rate at the fine-tuning stage is 0.005, which are also used in the original GG paper [17]. The initial decaying learning rate for SOM models is 0.1. The whole learning length of all models is set to 200 epochs and the predefined target unit number is set to 25. Different parameters have been tried and similar conclusions have been reached. To obtain more general results, each model is tested 10 times and the average results are used for comparison. Finally, a *t*-test is used to test whether or not the difference between the proposed CGG and the traditional GG models is of statistical significance.

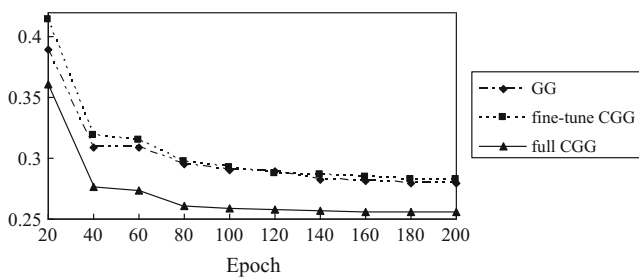


Fig. 5 A convergence comparison of models evaluated by AQE for a square data set

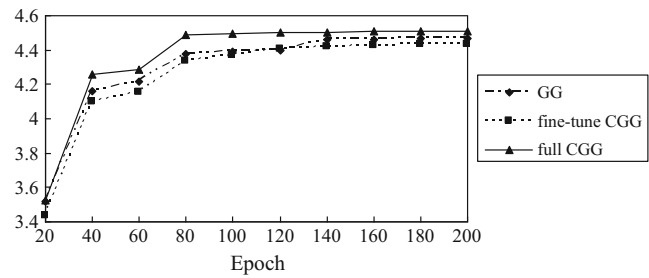


Fig. 6 A convergence comparison of models evaluated by error entropy for a square data set

7.1 The square data set

The square data set is used to demonstrate the ability of the SOM, constrained SOM, growing grid, fine-tune CGG, and full CGG to learn from a square data distribution. The square data set, proposed by Hung [23], uses 900 two-dimensional input samples from a 3×3 grid, which is between (0, 0) and (3, 3) at intervals of 0.1. Thus, this data set contains 900 evenly distributed points in a square. According to the experiments in Table 1, the traditional GG model outperforms the traditional SOM model, evaluated by criteria of average quantization error (AQE), error entropy, BMU activation rate (BAR) and average map scope (AMS). The static SOM model is better than the dynamic GG model, evaluated by the topographic error criterion. The constrained SOM is better than the traditional one. The fine-tune CGG model is better than the traditional GG model only in the criteria of BAR and AMS. One possible reason is that the effect of the constrained learning rule is limited when the learning rate is decayed to a very small value. However, if the constrained learning rule is used at both growing and fine-tuning stages, our proposed full CGG model outperforms GG and fine-tune CGG models.

We further use *t*-test to measure the degree of paired difference between GG models. Based on the criteria of AQE, BAR and AMS, the full CGG is significantly better than GG, which clearly indicates that our proposed model is an

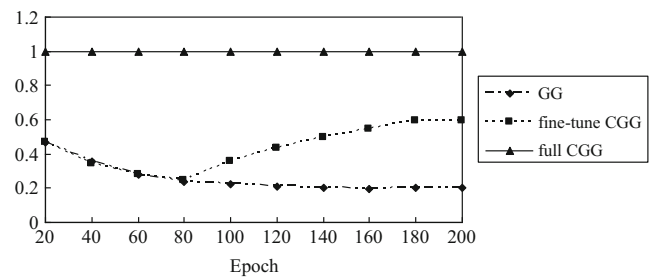


Fig. 7 A convergence comparison of models evaluated by BAR for a square data set

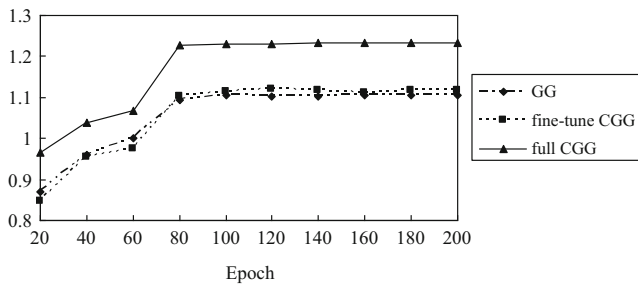


Fig. 8 A convergence comparison of models evaluated by AMS for a square data set

improvement on the traditional GG model on these criteria. In terms of the criteria of error entropy and TE, the full CGG model does not significantly outperform GG, the p -value in the t -test being greater than 0.05.

It is useful to evaluate different models by their convergence maps. The whole learning length of GG includes the growing and fine-tuning stages. Generally speaking, all convergence maps in Figs. 5, 6, 7, 8 and 9 are smooth due to taking the average results of 10 runs. In terms of the TE evaluation criterion (Fig. 9), the GG models appear to be slightly unstable during learning. However, the full CGG model performs better than other GG models in all evaluation criteria at each learning epoch.

7.2 The animal data set

The square data set contains input vectors which are clearly separated and thus the algorithms should have a good topological structure. The famous animal data set used in the literature [35] to demonstrate the ability of self-organizing semantic maps is also used in this paper. The animal data set contains 16 animals with 13 features and forms a 16 by 13 input matrix. Each row uses a schematic description of an animal, based on the presence (=1) or absence (=0) of some of the 13 different features. According to the experiments in Table 2 and Figs. 10, 11, 12, 13 and 14, the traditional GG model performs better than the traditional SOM model in all criteria except BAR. The fine-tune CGG model performs

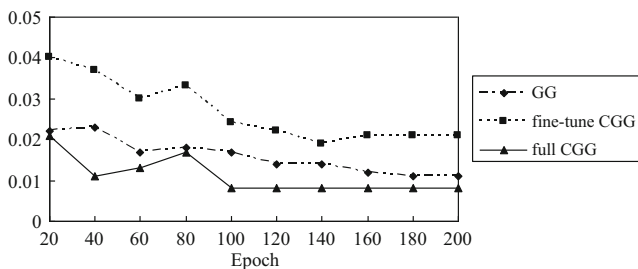


Fig. 9 A convergence comparison of models evaluated by TE for a square data set

Table 2 Performance Comparison Evaluated by AQE, error entropy, BAR, AMS and TE for an animal data set

Models	AQE	Error Entropy	BAR	AMS	TE
SOM	0.859	3.127	0.739	0.983	0.013
CSOM	0.818	3.134	1	1.009	0.044
GG	0.487	3.462	0.459	1.352	0
Fine-tune CGG	0.457** (0.000)	3.542** (0.000)	0.674** (0.000)	1.331 (0.109)	0.019 (0.193)
Full CGG	0.204** (0.000)	3.553 (0.167)	1** (0.000)	1.411** (0.000)	0.082** (0.000)

The value in bold form denotes a better performance than its benchmark. The p -value of t -test is shown in parenthesis

* denotes that the difference between two models achieves the 0.05 significance level

** denotes that the difference between two models achieves the 0.01 significance level

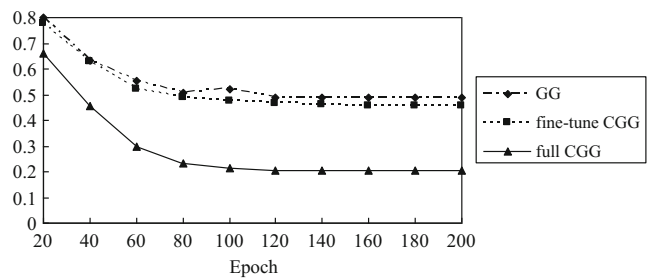


Fig. 10 A convergence comparison of models evaluated by AQE for an animal data set

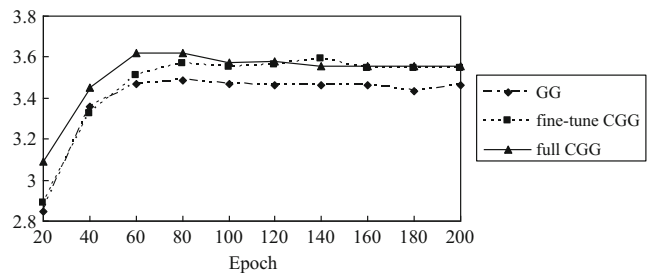


Fig. 11 A convergence comparison of models evaluated by error entropy for an animal data set

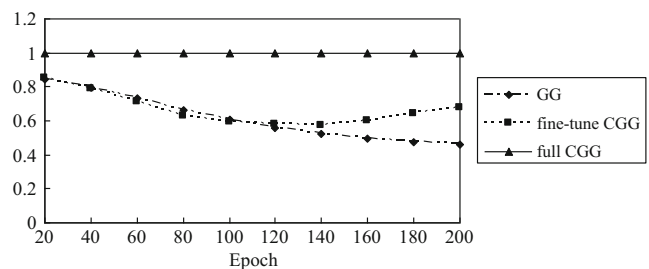


Fig. 12 A convergence comparison of models evaluated by BAR for an animal data set

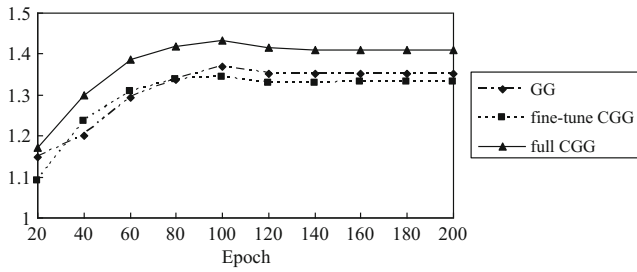


Fig. 13 A convergence comparison of models evaluated by AMS for an animal data set

significantly better than the traditional GG model, evaluated by the criteria of AQE, error entropy and BAR. The constrained SOM and full CGG models outperform their traditional counterparts in all criteria except TE. In terms of the TE criterion, all GG models show instability in the convergence process (Fig. 14). The constrained GG models do not perform better than the benchmark for this criterion.

7.3 The iris data set

The well-known iris data set collects iris flowers of three species, i.e. setosa, virginica and versicolor, each of which contains 50 input samples [36]. Each flower is illustrated by four features, i.e. the length of sepals, the length of petals, the width of sepals and the width of petals. According to the experiments in Table 3 and Figs. 15, 16, 17, 18 and 19, the traditional GG models perform better than the traditional SOM models in all criteria. The constrained SOM models perform better than the traditional SOM models in all criteria except TE. The fine-tune CGG models outperform the traditional GG models, evaluated by the criteria of AQE, error entropy and BAR. The full CGG models significantly outperform the traditional GG models in all criteria except TE.

7.4 The ionosphere data set

The ionosphere data set contains 351 radar samples with 34 continuous attributes and a class label, which builds a 351

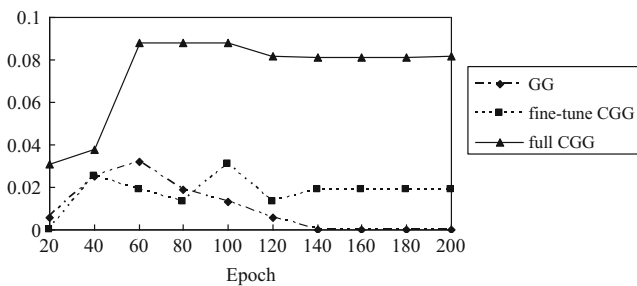


Fig. 14 A convergence comparison of models evaluated by TE for an animal data set

Table 3 Performance Comparison Evaluated by AQE, error entropy, BAR, AMS and TE for an iris data set

Models	AQE	Error Entropy	BAR	AMS	TE
SOM	0.454	3.652	0.127	1.494	0.233
CSOM	0.388	4.127	1	1.659	0.278
GG	0.409	4.174	0.370	1.949	0.093
Fine-tune CGG	0.371** (0.006)	4.293 (0.112)	0.475** (0.000)	1.859 (0.065)	0.102 (0.090)
Full CGG	0.337** (0.000)	4.483** (0.000)	1** (0.000)	1.987* (0.019)	0.166** (0.000)

The value in bold form denotes a better performance than its benchmark. The p-value of t-test is shown in parenthesis

*denotes that the difference between two models achieves the 0.05 significance level

**denotes that the difference between two models achieves the 0.01 significance level

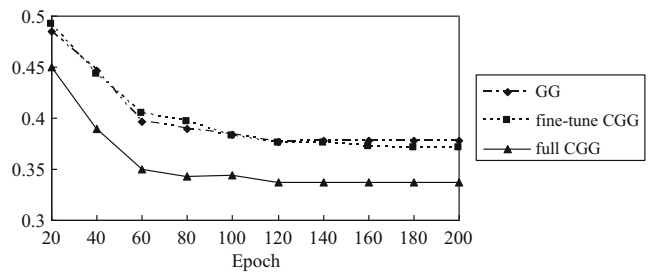


Fig. 15 A convergence comparison of models evaluated by AQE for an iris data set

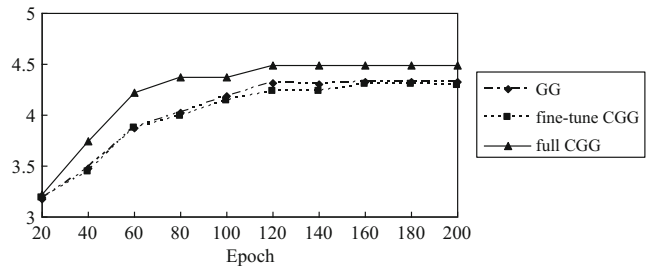


Fig. 16 A convergence comparison of models evaluated by error entropy for an iris data set

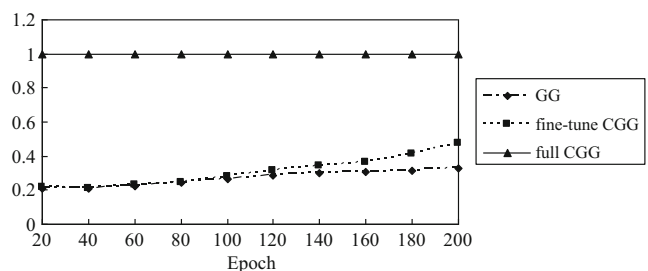


Fig. 17 A convergence comparison of models evaluated by BAR for an iris data set

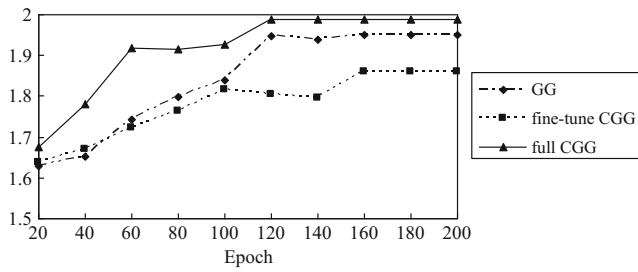


Fig. 18 A convergence comparison of models evaluated by AMS for an iris data set

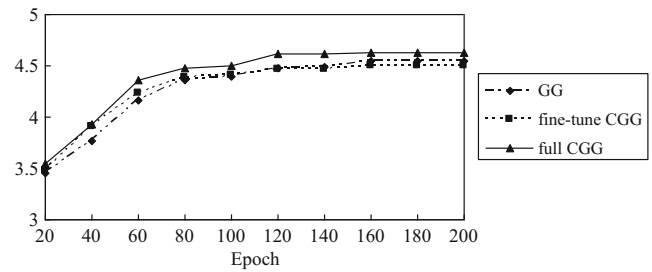


Fig. 21 A convergence comparison of models evaluated by error entropy for an ionosphere data set

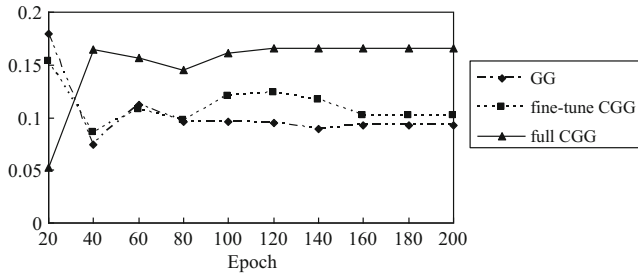


Fig. 19 A convergence comparison of models evaluated by TE for an iris data set

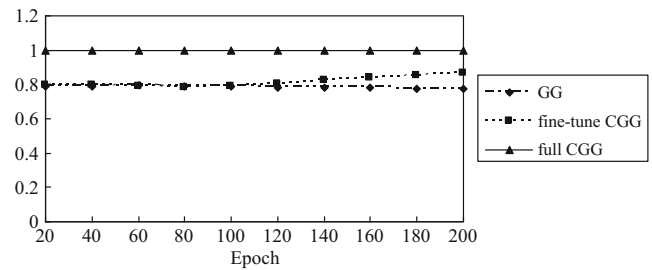


Fig. 22 A convergence comparison of models evaluated by BAR for an ionosphere data set

Table 4 Performance Comparison Evaluated by AQE, error entropy, BAR, AMS and TE for an ionosphere data set

Models	AQE	Error Entropy	BAR	AMS	TE
SOM	1.816	4.208	0.605	1.681	0.100
CSOM	1.769	4.429	1	1.804	0.083
GG	1.746	4.549	0.776	1.927	0.072
Fine-tune CGG	1.739 (0.162)	4.503** (0.004)	0.865** (0.000)	1.889* (0.014)	0.078 (0.512)
Full CGG	1.710** (0.000)	4.633** (0.003)	1** (0.000)	1.845** (0.004)	0.079 (0.135)

The value in bold form denotes a better performance than its benchmark. The p-value of t-test is shown in parenthesis

*denotes that the difference between two models achieves the 0.05 significance level

**denotes that the difference between two models achieves the 0.01 significance level

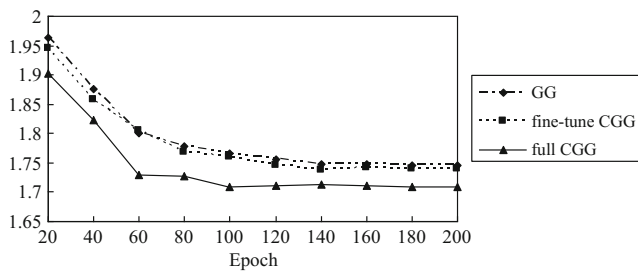


Fig. 20 A convergence comparison of models evaluated by AQE for an ionosphere data set

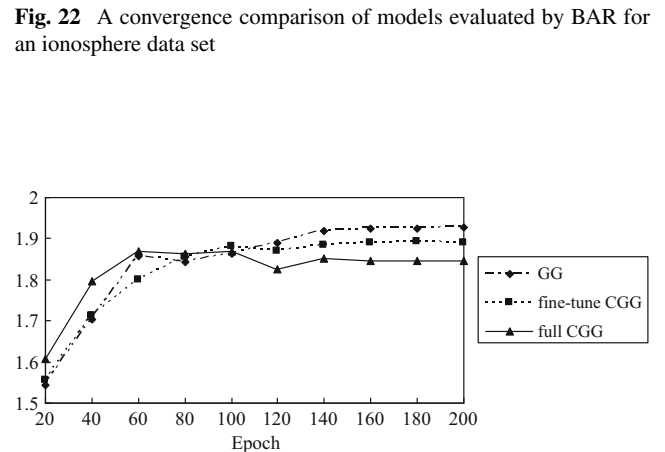


Fig. 23 A convergence comparison of models evaluated by AMS for an ionosphere data set

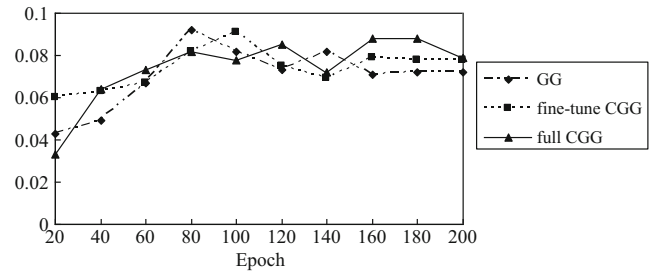


Fig. 24 A convergence comparison of models evaluated by TE for an ionosphere data set

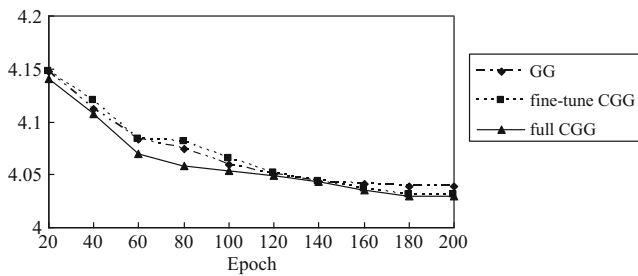


Fig. 25 A convergence comparison of models evaluated by AQE for a sentiment polarity data set

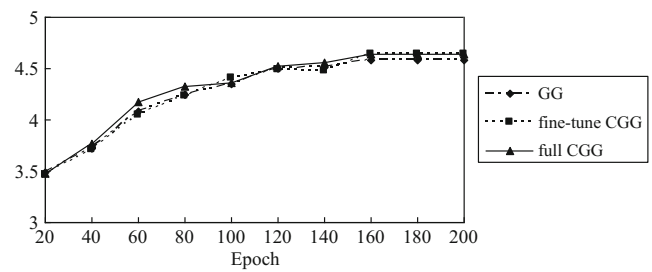


Fig. 26 A convergence comparison of models evaluated by error entropy for a sentiment polarity data set

by 34 input matrix [37]. Since the GG model is an unsupervised neural clustering approach, the class label is not included in this research. According to the experiments in Table 4 and Figs. 20, 21, 22, 23 and 24, the traditional GG models perform better than the traditional SOM models and the constrained SOM models perform better than the traditional SOM models, evaluated by all criteria. The full CGG models are significantly better than the traditional GG models for the criteria of AQE, error entropy and BAR, but worse for the criteria of AMS and TE. According to the convergence maps in Figs. 20, 21, 22, 23 and 24, the criterion of TE appear to be unstable during learning for all GG models.

7.5 Sentiment polarity data set

In order to test the proposed approaches in a high-dimensional data set, a well-known movie review data set is used. The sentiment polarity data set contains 2000 movie

Table 5 Performance Comparison evaluated by AQE, error entropy, BAR, AMS and TE for a sentiment polarity data set

Models	AQE	Error Entropy	BAR	AMS	TE
SOM	4.014	4.459	0.999	2.151	0.246
CSOM	4.007	4.464	1	2.183	0.240
GG	4.039	4.583	0.999	2.616	0.116
Fine-tune CGG	4.031* (0.015)	4.637* (0.027)	0.999 (0.193)	2.565 (0.076)	0.103* (0.044)
Full CGG	4.030 (0.063)	4.636** (0.000)	1* (0.15)	2.657 (0.322)	0.110 (0.108)

The value in bold form denotes a better performance than its benchmark. The p-value of t-test is shown in parenthesis

*denotes that the difference between two models achieves the 0.05 significance level

**denotes that the difference between two models achieves the 0.01 significance level

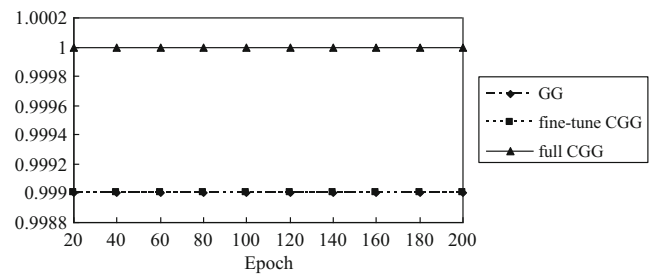


Fig. 27 A convergence comparison of models evaluated by BAR for a sentiment polarity data set

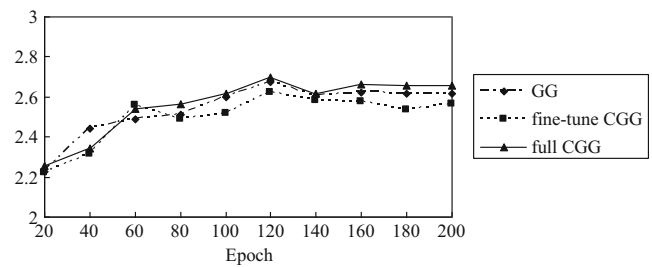


Fig. 28 A convergence comparison of models evaluated by AMS for a sentiment polarity data set

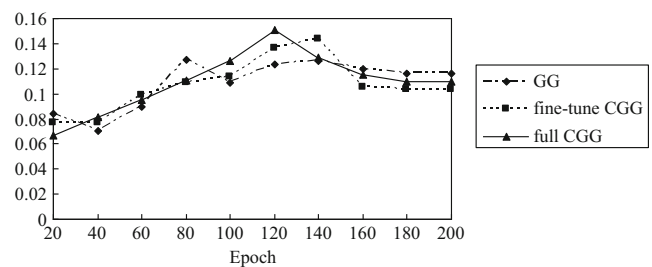


Fig. 29 A convergence comparison of models evaluated by TE for a sentiment polarity data set

Table 6 Description of fundamental clustering problem suite (FCPS)

	Atom	Chain Link	Engy Time	Golf Ball	Hepta	Lsun	Target	Tetra	Wing Nut	Two Diamonds
Samples	800	1000	4096	4002	212	400	770	400	1070	800
Attributes	3	3	2	3	3	2	2	3	2	2

Table 7 Performance Comparison Evaluated by AQE for FCPS

Models	Atom	Chain Link	Engy Time	Golf Ball	Hepta	Lsun	Target	Tetra	Wing Nut	Two Diamonds
GG	13.444	0.492	0.523	0.364	0.672	0.317	0.334	0.534	0.319	0.205
Fine-tune	13.190	0.481**	0.505**	0.352**	0.651**	0.320	0.336	0.536	0.314	0.203
CGG	(0.091)	(0.008)	(0.001)	(0.000)	(0.002)	(0.053)	(0.511)	(0.107)	(0.062)	(0.421)
Full CGG	10.616**	0.346**	0.474**	0.316**	0.502**	0.268**	0.245**	0.488**	0.270**	0.169**
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

The value in bold form denotes a better performance than its benchmark. The p-value of t-test is shown in parenthesis

* denotes that the difference between two models achieves the 0.05 significance level

** denotes that the difference between two models achieves the 0.01 significance level

Table 8 Performance Comparison Evaluated by Error Entropy for FCPS

Models	Atom	Chain Link	Engy Time	Golf Ball	Hepta	Lsun	Target	Tetra	Wing Nut	Two Diamonds
GG	4.257	4.237	4.482	4.731	3.371	4.204	4.104	4.441	4.488	4.420
Fine-tune	4.249	4.253	4.529*	4.761*	3.444**	4.175*	4.109	4.447	4.495	4.423
CGG	(0.875)	(0.552)	(0.027)	(0.027)	(0.013)	(0.037)	(0.623)	(0.734)	(0.547)	(0.838)
Full CGG	4.563**	4.359*	4.624**	4.748	4.200**	4.503**	4.395**	4.588**	4.711**	4.654**
	(0.000)	(0.014)	(0.000)	(0.391)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

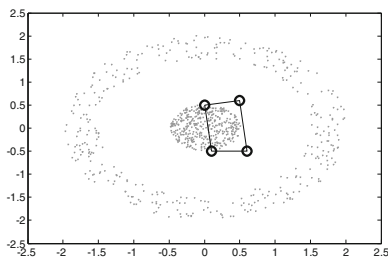
Table 9 Performance Comparison Evaluated by BAR for FCPS

Models	Atom	Chain Link	Engy Time	Golf Ball	Hepta	Lsun	Target	Tetra	Wing Nut	Two Diamonds
GG	0.412	0.243	0.226	0.378	0.194	0.147	0.186	0.463	0.196	0.203
Fine-tune	0.620**	0.620**	0.502**	0.621**	0.517**	0.491**	0.505**	0.684**	0.535**	0.448**
CGG	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.001)
Full CGG	1**	1**	1**	1**	1**	1**	1**	1**	1**	1**
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

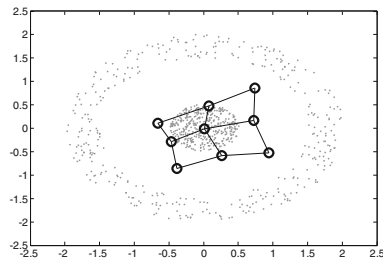
Table 10 Performance Comparison Evaluated by AMS for FCPS

Models	Atom	Chain Link	Engy Time	Golf Ball	Hepta	Lsun	Target	Tetra	Wing Nut	Two Diamonds
GG	24.823	1.387	1.873	0.774	2.244	1.485	1.025	1.214	1.192	1.041
Fine-tune	26.552	1.337*	1.794**	0.793**	2.183*	1.468	1.004	1.219	1.189	1.068*
CGG	(0.241)	(0.016)	(0.003)	(0.000)	(0.020)	(0.185)	(0.798)	(0.547)	(0.897)	(0.015)
Full CGG	34.104**	1.513**	2.131**	0.827**	2.502**	1.638**	1.620**	1.287**	1.292**	1.123**
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

(a) 2x2 grid



(b) 3x3 grid



(c) 5x5 grid

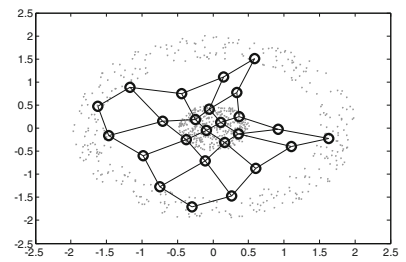
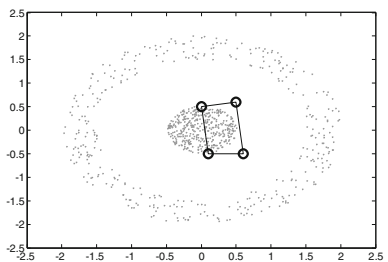
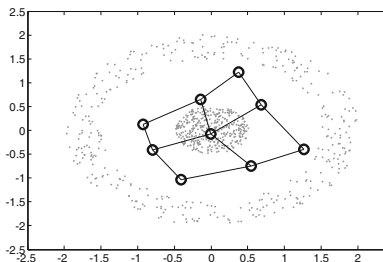


Fig. 30 The convergence maps of growing grid after 0, 20 and 200 epochs for the target data set. A circle denotes an output unit and a grey dot denotes an input vector

(a) 2x2 grid



(b) 3x3 grid



(c) 6x4 grid

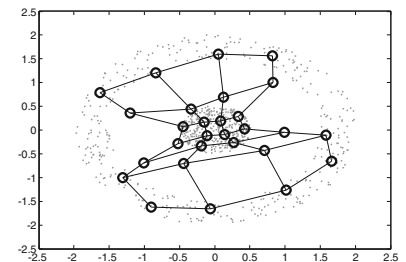
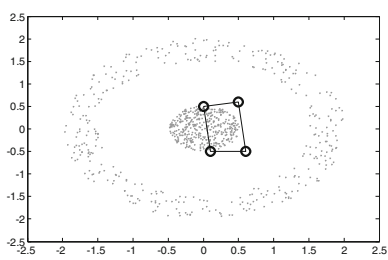
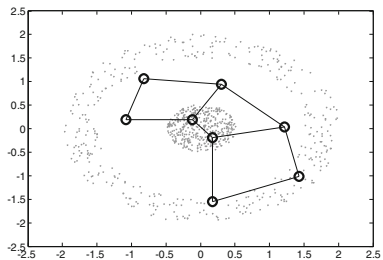


Fig. 31 The convergence maps of fine-tune CGG after 0, 20 and 200 epochs for the target data set. A circle denotes an output unit and a grey dot denotes an input vector

(a) 2x2 grid



(b) 4x2 grid



(c) 5x5 grid

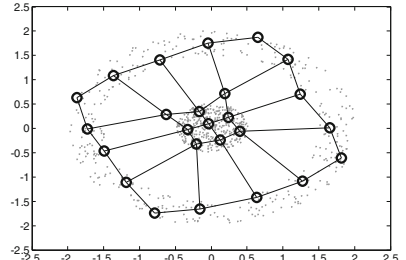


Fig. 32 The convergence maps of full CGG after 0, 20 and 200 epochs for the target data set. A circle denotes an output unit and a grey dot denotes an input vector

Table 11 Performance Comparison Evaluated by TE for FCPS

Models	Atom	Chain Link	Engy Time	Golf Ball	Hepta	Lsun	Target	Tetra	Wing Nut	Two Diamonds
GG	0.109	0.081	0.136	0.080	0.037	0.200	0.082	0.084	0.044	0.052
Fine-tune	0.099	0.087	0.179**	0.084	0.042	0.169*	0.088	0.100**	0.054**	0.055
CGG	(0.301)	(0.153)	(0.000)	(0.424)	(0.455)	(0.014)	(0.312)	(0.009)	(0.008)	(0.312)
Full CGG	0.095*	0.079	0.096**	0.100**	0.088**	0.180	0.029**	0.069*	0.040*	0.071**
	(0.031)	(0.348)	(0.004)	(0.002)	(0.000)	(0.060)	(0.000)	(0.046)	(0.011)	(0.000)

reviews extracted from Internet movie database (IMDb, <http://www.imdb.com/>) by Pang and Lee [38]. After text processing in the work of Hung and Lin [39], this data set contains 3131 attributes and builds a 2000 by 3131 input matrix. According to the experiments in Table 5 and Figs. 25, 26, 27, 28 and 29, the constrained SOM models outperform traditional SOM models for all evaluation criteria. The traditional GG models perform better than the traditional SOM models, evaluated by the criteria of error entropy, AMS and TE. The fine-tune CGG models perform significantly better than the traditional GG models, evaluated by the criteria of AQE, error entropy and TE. The full CGG models outperform the traditional GG models in all evaluation criteria.

7.6 Fundamental clustering problem suite (FCPS)

The fundamental clustering problem suite (FCPS) contains ten data sets, which are treated as test beds for neural clustering [40]. A short description of data sets is shown in Table 6. Based on the criteria of AQE, error entropy and BAR, except in a few cases, the experiments in Tables 7, 8 and 9 demonstrate that the fine-tune CGG models perform better than the traditional GG models. According to the experiments shown in Tables 7, 8, 9 and 10, the full CGG models significantly outperform the traditional GG models, evaluated by the criteria of AQE, error entropy, BAR and AMS. In terms of the TE criterion, the full CGG models perform better than the traditional GG models for seven datasets, which are Atom, Chain Link, Engy Time, Lsun, Target, Tetra and Wing Nut. Thus, the experiments conclude that the proposed full CGG models can significantly improve the traditional GG model based on five evaluation criteria. Convergence maps of the traditional GG, fine-tune CGG and full CGG are shown in Figs. 30, 31 and 32, respectively. From these convergence maps, we can evaluate the quality of growing maps based on visualization, and show that the proposed full CGG is more effective than the others (Table 11).

8 Conclusion

The Growing Grid (GG) is a very useful algorithm for mapping a high-dimensional data set onto a low-dimensional topological map without using the predefined topological structure before learning. It enforces a non-linear projection, a non-parametric statistical technique and an unsupervised competitive learning rule. In this paper, a novel constrained GG (CGG) is proposed, which emphasizes the effect on the lateral connections of output units and neutralizes the effect of the input vectors on the neighbors. The fine-tune CGG uses the constrained neural learning rule only at its fine-tuning stage, whereas the full CGG uses the constrained neural learning rule at both growing and fine-tuning stages. Fifteen data sets are used and evaluated by five criteria, which are average quantization error (AQE), error entropy, BMU activation rate (BAR), average map scope (AMS) and topographic error (TE). We compare the performance of the traditional SOM, constrained SOM, GG, fine-tune CGG and full CGG based on the average results from ten experiments for each evaluation criterion. Except in some cases, the performance of the traditional GG is improved if the constrained learning rule is used, and the constrained neural learning rule used at both growing and fine-tuning stages outperforms only at the fine-tuning stage. Finally, we use the *t*-test to test whether the improvements from our proposed model achieve statistical significance or not. Except in some cases, our proposed full CGG model significantly outperforms those models with an original learning rule. These exceptions may occur due to some GG models randomly building a twisted map at their growing stage, so that the fine-tune CGG or full CGG cannot produce a significant improvement.

9 Scope for future work

For future work, the following issues may be considered. Firstly, our proposed approach could be applied to other

competitive learning models, such as neural gas (NG), growing neural gas (GNG), growing neural gas with utility (GNG-U), grow when required (GWR), dynamic adaptive self-organizing hybrid (DASH), etc. Secondly, more evaluation and application methods may be considered. For example, the receiver operating characteristics (ROC) curve could be explored as a recommender system, and other data sets from different domains may also be utilized for comparison. Thirdly, another important issue is how to avoid building a twisted topological map. This may occur due to some uncontrollable random parameters. However, such random parameters should not simply be removed, i.e. initial random unit weights and random input sample order. A twist-free mechanism would be a welcome addition in the field, and is a possible direction for further research. Fourthly, according to some experiments, it seems difficult to improve the performance of topological preservation based on the topographic error (TE) criterion for neural growing models. The possible reason may be inadequate learning length for each growing process. As this research only focuses on the principle of winner-take-more, further research could evaluate the relationship between the learning length and effectiveness of topological preservation. Fifthly, typical mapping algorithms have online and offline specifications of algorithms. However, this research applies only to online growing grid algorithms. In future work, the proposed constrained neural learning could be used in offline mapping algorithms and it should be interesting to compare the difference between online and offline constrained models.

References

- Kohonen T (1982) Self-organized formation of topologically correct feature maps. *Biol Cybern* 43:59–69
- Hosseini SY, Bideh AZ (2014) A data mining approach for segmentation-based importance-performance analysis (SOM-BPNN-IPA): a new framework for developing customer retention strategies. *Sev Bus* 8(2):295–312
- Kohonen T (2001) *Self-organizing maps*. Springer-Verlag, New York
- Bohlooli A, Jamshidi K (2012) A GPS-free method for vehicle future movement directions prediction using SOM for VANET. *Appl Intell* 36(3):685–697
- Hung C, Tsai C-F (2008) Market segmentation based on hierarchical self-organizing map for markets of multimedia on demand. *Expert Syst Appl* 34:780–787
- do Rêgo RLME, Araújo AFR (2010) de Lima Neto FB. Growing self-reconstruction maps. *IEEE Trans Neural Netw* 21(2):211–213
- Linda O, Manic M (2011) Self-organizing fuzzy haptic teleoperation of mobile robot using sparse sonar data. *IEEE Trans Ind Electron* 58(8):3187–3195
- Kamimura R (2011) Structural enhanced information and its application to improved visualization of self-organizing maps. *Appl Intell* 34(1):102–115
- Chihhi H, Chainbi W, Ghedira K (2013) An energy-efficient self-provisioning approach for cloud resources management. *ACM SIGOPS Oper Syst Rev* 47(3):2–9
- Fritzke B (1994) Growing cell structures—a self-organizing network for unsupervised and supervised learning. *Neural Netw* 7(9):1441–1460
- Fritzke B (1995) A growing neural gas network learns topologies. In: Tesauro G, Touretzky DS, Leen TK (eds) *Advances in neural information processing systems*, vol 7. MIT Press, Cambridge MA, pp 625–632
- Fritzke B (1997) A self-organizing network that can follow non-stationary distributions. In: *International conference on artificial neural networks*, pp 613–618
- Marsland S, Shapiro J, Nehmzow U (2002) A self-organising network that grows when required. *Neural Netw* 15:1041–1058
- Hung C, Wermter S (2003) A dynamic adaptive self-organising hybrid model for text clustering. In: *Third IEEE international conference on data mining*, pp 75–82
- Furao S, Hasegawa O (2006) An incremental network for on-line unsupervised classification and topology learning. *Neural Netw* 19(1):90–106
- Araújo AFR, Costa DC (2009) Local adaptive receptive field self-organizing map for image color segmentation. *Image and Vis Comput* 27(9):1229–1239
- Fritzke B (1995) Growing grid—a self-organizing network with constant neighborhood range and adaptation strength. *Neural Process Lett* 2(5):9–13
- Kohonen T (1984) In: *Self-organization and associative memory*. Springer-Verlag, Berlin
- Grossberg S (1987) Competitive learning: from interactive activation to adaptive resonance. *Cogn Sci* 11:23–63
- Martinetz T, Schulten KA (1991) Neural-Gas network learns topologies. *Artifi Neural Netw* 1:397–402
- Martinetz TM (1993) Competitive Hebbian learning rule forms perfectly topology preserving maps. In: *International conference on artificial neural networks*, pp 427–434
- Araújo AFR, do Rêgo RLME (2013) Self-organizing maps with a time-varying structure. *ACM Comput Surv* 46(1):7
- Hung C (2008) A constrained neural learning rule for eliminating the border effect in online self-organising maps. *Connect Sci* 20(1):1–20
- Kohonen T (1993) Physiological interpretation of the self-organizing map algorithm. *Neural Netw* 6:895–905
- Lo Z-P, Bavarian B (1991) On the rate of convergence in topology preserving neural networks. *Biol Cybern* 65:55–63
- Van Hulle MM (2000) In: *Faithful representations and topographic maps: from distortion to information-based self-organization*. Wiley-Interscience, New York
- Göppert J, Rosenstiel W (1994) Dynamic extensions of self-organizing maps. In: *International conference on artificial neural networks*, pp 330–333
- Wu Y, Takatsuka M (2005) The geodesic self-organizing map and its error analysis. In: *Twenty-eighth Australasian conference on computer science*, pp 343–351
- Kiviluoto K (1996) Topology preservation in self-organizing maps. In: *International conference on neural networks*, pp 294–299
- Shannon CE, Weaver W (1949) In: *The mathematical theory of communication*. University of Illinois Press, Urbana
- Khalilia M, Popescu M (2014) *Topology preservation in fuzzy self-organizing maps*, vol 312, pp 105–114

32. Bauer H-U, Pawelzik KR (1992) Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Trans Neural Netw* 3(4):570–578
33. Villmann T, Der R, Hermann M, Martinetz T (1997) Topology preservation in self-organizing feature maps: exact definition and measurement. *IEEE Trans Neural Netw* 8(2):256–266
34. Uriarte EA, Martin FD (2008) Topology preservation in SOM. *World Academy of Science. Eng Technol* 2:867–870
35. Rauber A, Paralic J, Pampalk E (2000) Empirical evaluation of clustering algorithms. *J Inf Organ Sci* 24(2):195–209
36. Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Annu Eugen* 7:178–188
37. Sigillito VG, Wing SP, Hutton LV, Baker KB (1989) Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Tech Dig* 10:262–266
38. Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: the 42nd Annual Meeting of the Association for Computational Linguistics, pp 271–278
39. Hung C, Lin H-K (2013) Using objective words in SentiWordNet to improve word-of-mouth sentiment classification. *IEEE Intell Syst* 28(2):47–54
40. Ultsch A (2005) Clustering with SOM: U*C. In: Workshop on Self-Organizing Maps, pp 75–82



Chihli Hung received a PhD at School of Computing and Technology from the University of Sunderland, UK in 2004. He is an associate professor at the Department of Information Management, Chung Yuan Christian University, Taiwan, R.O.C. His current research interests are in text mining, intelligence systems, computational intelligence, and data mining.