Neural Comput & Applic (2004) 13: 168–174
DOI 10.1007/s00521-004-0419-y

ORIGINAL ARTICLE

L. D. Alahakoon

Controlling the spread of dynamic self-organising maps

Received: 7 April 2004 / Accepted: 20 April 2004 / Published online: 25 May 2004

Abstract The growing self-organising map (GSOM) has recently been proposed as an alternative neural network architecture based on the traditional self-organising map (SOM). The GSOM provides the user with the ability to control the spread of the map by defining a parameter called the spread factor (SF), which results in enhanced data mining and hierarchical clustering opportunities. When experimenting with the SOM, the grid size (number of rows and columns of nodes) can be changed until a suitable cluster distribution is achieved. In this paper we highlight the effect of the spread factor on the GSOM and contrast this effect with grid size change (increase and decrease) in the SOM. We also present experimental results in support of our claims regarding differences between GSOM and SOM.

1 Introduction

The growing self-organising map (GSOM) has been proposed as a dynamically generating neural map which has particular advantages for data mining applications [1, 4]. Several researchers have previously developed incrementally growing SOM models [5, 6, 8]. These models have similarities to and differences from each other, but they all attempt to solve the problem of pre-defined, fixed structure SOMs. A parameter called the spread factor (SF) in the GSOM provides the data analyst with control over the spread of the map. The ability to control the spread of the map is unique to the GSOM and could be manipulated by the data analyst to achieve progressive clustering of a data set at different levels of detail. The SF can be assigned values in the range 0 to 1 which are independent of the dimensionality of the data set. Thus the SF provides a measure of the *level of spread* across different data sets. The spread of the GSOM can be increased by using a higher SF value. Such spreading can continue until the analyst is satisfied with the *level of clustering* achieved or until each node is identified as a separate cluster.

A data analyst using the SOM experiments with different map sizes by mapping the data set to grids (generally two-dimensional) of different sizes. In this paper we highlight the difference between such grid size change in traditional SOM usage with controlled map spreading in the GSOM. We describe the SF-controlled *growth* of the GSOM as a technique for more representative feature-map generation where the inter and intra-cluster relationships are better visualised. The fixed structure SOM forces distortion on its mapping because of its inability to expand when required.

The traditional SOM does not provide a measure for identifying the size of a feature map with it's *level of spread*. Therefore the data analyst using the SOM can only refer to the length and width of the grid to relate to map size. We show that the shape or size of the SOM cannot be meaningfully related to the spread of the data without accurate knowledge of the distribution of the data set. The use of the SF in the GSOM provides a single indicator with which to dictate map spread and also the ability to relate a map to a particular level of spread.

The identification of clusters in a map will always depend on the data set being used and on the *needs* of the application. For example, the analyst may only be interested in identifying the most significant clusters for one application. Alternatively, the analyst may become interested in more detailed clusters. Therefore the clusters identified will depend on the *level of significance* required by the analyst at a certain instance in time. In data mining-type applications, because the analyst is not aware of the clusters in the data, it is generally necessary to study the clusters generated at different levels of spread (detail) to obtain an understanding of the data. Attempting to obtain different sized SOMs for this purpose can result in distorted views due to the need to force a data set into maps with pre-defined structure.

L. D. Alahakoon
School of Business Systems,
Monash University, Clayton,
Victoria, Australia
E-mail: damminda.alahakoon@infotech.monash.edu.au

Therefore we describe the number of clusters shown in a map also as a variable which depends on the level of significance required by the data analyst. We present the SF as our technique for providing the level of significance as a parameter in the map-generation process.

In Sect. 2 we provide a description of the GSOM and SF techniques. Section 3 explains the workings of the spreadout effect in the GSOM with the SF as a parameter. Here we also compare the GSOM and SOM in terms of the spreading-out effect. In Sect. 4 we describe experimental results to clarify our claims. Section 5 provides the conclusions to the paper.

2 Controlling the GSOM spread with the spread factor

2.1 Growing self-organising map

The GSOM is an unsupervised neural network which is initialised with four nodes and *grows* nodes to represent the input data [2, 3]. During the node growth, the weight values of the nodes are *self organised* according to a similar method as the SOM.

The GSOM process is as follows:

1. Initialisation phase
   (a) Initialise the weight vectors of the starting nodes (4) with random numbers.
   (b) Calculate the growth threshold (GT) for the given data set according to the user requirements.
2. Growing phase
   (a) Present input to the network.
   (b) Determine the weight vector that is closest to the input vector mapped to the current feature map (winner), using Euclidean distance (similar to the SOM). This step can be summarised as:
   Find q' such that $c|v - w_{q'}| \leq |v - w_q| \; \forall q \in \mathcal{N}$ where v, w are the input and weight vectors, respectively, q is the position vector for nodes, and $\mathcal{N}$ is the set of natural numbers.
   (c) The weight vector adaptation is applied only to the neighbourhood of the winner and to the winner itself. The neighbourhood is a set of neurons around the winner, but in the GSOM the starting neighbourhood selected for weight adaptation is smaller than the SOM (localised weight adaptation). The amount of adaptation (learning rate) is also reduced exponentially over the iterations. Even within the neighbourhood weights which are closer to the winner are adapted more than those further away. The weight adaptation can be described by:

$$w_j(k+1) = \begin{cases} w_j(k), & j \notin N_{k+1} \\ w_j(k) + LR(k)(x_k - w_j(k)), & j \in N_{k+1} \end{cases}$$

   where the learning rate $LR(k)$, $k \in \mathcal{N}$ is a sequence of positive parameters converging to zero as $k \to \infty$. $w_j(k)$, $w_j(k+1)$ are the weight vectors of node j, before and after the adaptation, and N_{k+1} is the neighbourhood of the winning neuron at $(k+1)$th iteration. The decreasing of $LR(k)$ in the GSOM, depends on the number of nodes existing in the network at time k.
   (d) Increase the error value of the winner (error value is the difference between the input vector and the weight vectors).
   (e) When $TE_i \geq GT$ (where TE is the total error of node i and GT is the growth threshold), grow nodes if i is a boundary node and distribute weights to neighbours if i is a non-boundary node.
   (f) Initialise the new node-weight vectors to match the neighbouring node weights.
   (g) Initialise the learning rate (LR) to its starting value.
   (h) Repeat steps b to g until all inputs have been presented, and node growth is reduced to a minimum level.
3. Smoothing phase
   (a) Reduce learning rate and fix a small starting neighbourhood.
   (b) Find winner and adapt weights of winner and neighbours in the same way as in the growing phase. Therefore, instead of the weight adaptation in the original SOM, the GSOM adapts its weights and architecture to represent the input data. Therefore in the GSOM a node has a weight vector and two-dimensional coordinates which identify its position in the net, whereas in the SOM the weight vector is also called the position vector.

2.2 The spread factor

As described in the algorithm, the GSOM uses a threshold value called the GT to decide when to initiate new node growth. GT will decide the amount of spread of the feature map to be generated. Therefore if we require only a very abstract picture of the data, a large GT will result in a map with fewer nodes. Similarly a smaller GT will result in the map spreading out more. When using the GSOM for data mining, it might be a good idea to first generate a *smaller* map, showing the most significant clustering in the data only, which will give the data analyst a summarised picture of the inherent clustering in the total data set.

The node growth in the GSOM is initiated when the error value of a node exceeds the GT. The total error value for node i is calculated as:

$$TE_i = \sum_{H_i} \sum_{j=1}^{\mathscr{D}} (x_{i,j} - w_j)^2 \quad (1)$$

where H is the number of hits to the node i and $\mathscr{D}$ is the dimension of the data. $x_{i,j}$ and w_j are the input and weight vectors of the node i respectively. For new node growth:

$$TE_i \geq GT \quad (2)$$

The GT value has to be experimentally decided depending on our requirement for map growth. As can be seen from Eq. 1 the dimension of the data set will make a significant impact on the accumulated error (TE) value, and as such will have to be considered when deciding the GT for a given application. The SF was introduced to address this limitation, thus eliminating the need for the data analyst to consider data dimensionality when generating feature maps. Although exact measurement of the effect of the SF is yet to be carried out, it provides an indicator for identifying the level of spread in a feature map (across different dimensionalities). The derivation of the SF has been described elsewhere [4]. The formula used is:

$$GT = -D \times \ln(SF)$$

Therefore, instead of having to provide a GT, which would take different values for different data sets, the data analyst has to provide a value SF, which will be used by the system to calculate the GT value depending on the dimensionality of the data. This will allow the GSOMs to be identified with their spread factors, and will be a basis for comparison of different maps.

3 The spreading-out effect of the GSOM compared with the traditional SOM

3.1 The relationship between the shape of the SOM and the input data distribution

Figure 1 shows the diagram of a SOM with four clusters A, B, C, and D which can be used to explain the spread of clusters due to the change of grid size in a SOM. As shown in Fig. 1a the SOM has a grid of length and width X and Y, respectively. The intra-cluster distances are x and y as shown in Fig. 1a. In Fig. 1b a SOM has been generated on the same data but the length of the grid has been increased (to $Y' > Y$) while the width has been maintained at the previous value ($X = X'$). The intra-cluster distances in Fig. 1b are x' and y'. It can be seen that inter-cluster distances in the y direction have changed in such a way that the cluster positions have been *forced* into maintaining the proportions of the SOM grid. The clusters themselves have been *dragged out* in the y direction due to the intra-cluster distances also being forced by the grid. Therefore in Fig. 1 $X{:}Y \approx x{:}y$ and $X'{:}Y' \approx x'{:}y'$. This phenomenon can be considered in an intuitive manner as follows [7]:

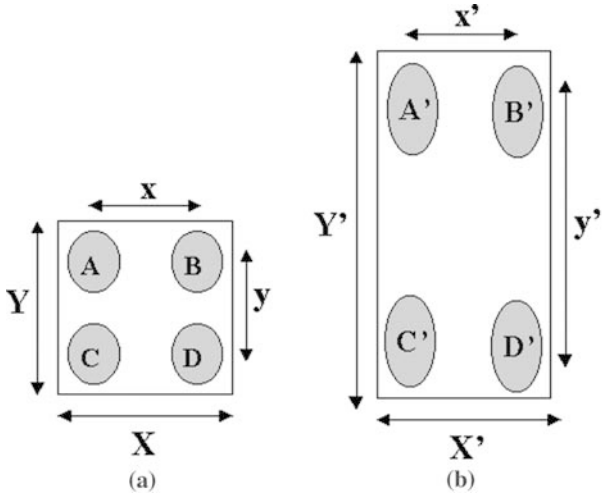

Fig. 1 The shift of the clusters on a feature map due to the shape and size

Considering two dimensional maps, the inter and intra-cluster distances in the map can be separately identified in the X and Y directions. We simply visualise the spreading-out effect of the SOM as the inter and intra-cluster distances in the X and Y directions proportionally being adjusted to fit in with the width and length of the SOM.

The same effect has been described by Kohonen [7] as a limitation of the SOM called the *oblique orientation*. This limitation has been observed and demonstrated experimentally with a two-dimensional grid, and we use the same experiment to indicate the limitations of the SOM for data mining.

Figure 2a shows a 4×4 SOM for a set of artificial data selected from a uniformly distributed two-dimensional square region. The attribute values x, y in the data are selected such that $x{:}y = 4{:}4$ and as such the grid in Fig. 2a is well spread out, providing an optimal map. In Fig. 2b the input value attributes $x{:}y \neq 4{:}4$ while the input data *demands* a grid of 4:4 or similar proportions. As such it has resulted in a distorted map with a crushed effect. Kohonen has described oblique orientation as resulting from significant differences in variance of the components (attributes) of the input data. Therefore the grid size of the feature map has to be initialised to match the values of the data attributes or dimensions to obtain a *properly* spread out map. For example, consider a two dimensional data set where the attribute values have the proportion $x{:}y$. In such an instance a two-dimensional grid can be initialised with $n \times m$ nodes where $n{:}m = x{:}y$. Such a feature map will produce an optimal spread of clusters maintaining the proportionality in the data. But in many data-mining applications the data analyst is not aware of the data attribute proportions. Also the data are mostly of very high dimensions, and as such it becomes impossible to decide a suitable two-dimensional grid structure and shape. Therefore initialing with an optimal grid for SOM becomes a non-feasible solution.

Kohonen has suggested a solution to this problem by introducing *adaptive tensorial weights* in calculating the distance for identifying the winning nodes in the SOM during training. The formula for distance calculation is:

$$d^2[x(t), w_i(t)] = \sum_{j=1}^{N} \psi_{i,j}^2 [\xi_j(t) - \mu_{i,j}(t)]^2 \quad (3)$$

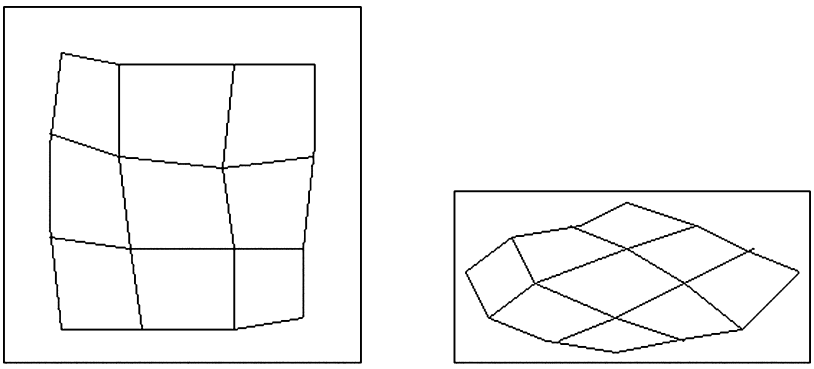

Fig. 2 Oblique orientation of an SOM

where ξ_j are the attributes (dimensions) of input x, the $\mu_{i,j}$ are the attributes of w_i, and $\psi_{i,j}$ is the weight of the jth attribute associated with node i. The values of $W_{i,j}$ are estimated recursively during the unsupervised learning process [7]. The resulting adjustment has been demonstrated using artificial data sets in Fig. 3.

The variance of the input data along the vertical dimension (attribute) versus the horizontal one is varied (1:1, 1:2, 1:3, and 1:4 in Figs. 3a–d, respectively). The results for the unweighted and weighted maps are shown on the left and right, respectively, in Fig. 3.

We interpret the *oblique orientation* as an occurrence due to the map attempting to fit in with a pre-defined network, and resulting in a distorted structure. The tensorial weights method attempts to reduce the oblique orientation while still keeping within the network *borders*, thus forcing the shape of the network on the data. This is opposite to the ideal solution, because it is the data which should dictate the size and shape of the grid. By changing the size of the grid in the SOM, the map is forced to fit in with a new network size and shape. If the data attributes are not proportionate (in the x and y directions) to the network grid, a distorted final map can occur.

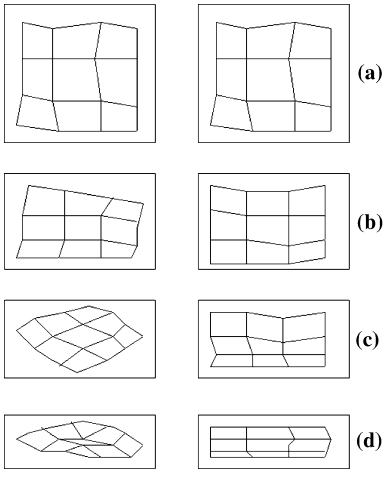

Fig. 3 Solving oblique orientation with tensorial weights (from Ref. [7])

3.2 Effect of the spread factor on the GSOM

In GSOM, the map is spread out by using different SF values. According to the formula presented in Sect. 2, a *low* SF value will result in a *higher* GT. In such a case a node will accommodate a higher error value before it initiates a growth. Therefore we can state the spreading-out effect (or new node generation) of GSOM as follows.

The criterion for new node generation from node i in the GSOM is:

$$E_{i,\text{tot}} \geq \text{GT} \tag{4}$$

where $E_{i,\text{tot}}$ is the total accumulated error of node i and GT is the growth threshold. The $E_{i,\text{tot}}$ is expressed as:

$$E_{i,\text{tot}} = \sum_{H_i} \sum_{j=1}^{D} (x_j(t) - w_j(t))^2 \tag{5}$$

If we denote low SF and high SF values by SF_{low} and SF_{high}, respectively, and $\Rightarrow$ denotes *implies*, then:

$$\text{SF}_{\text{low}} \Rightarrow \text{GT}_{\text{high}} \tag{6}$$

$$\text{SF}_{\text{high}} \Rightarrow \text{GT}_{\text{low}} \tag{7}$$

Therefore from Eqs. 5, 6 and 7 we can say that when the $\text{SF} = \text{SF}_{\text{low}}$, node i will generate new nodes when:

$$E_{i,\text{tot}} = \underbrace{\sum_{H_i} \sum_{j=1}^{D} (x_j(t) - w_j(t))^2}_{R_l} \geqslant \text{GT}_{\text{high}} \tag{8}$$

Similarly, when $\text{SF} = \text{SF}_{\text{high}}$, node i will generate new nodes when:

$$E_{i,\text{tot}} = \underbrace{\sum_{H_i} \sum_{k=1}^{D} (x_k(t) - w_k(t))^2}_{R_s} \geqslant \text{GT}_{\text{low}} \tag{9}$$

where $x_j \in R_l$ and $x_k \in R_s$ are two regions in the input data space.

It can be seen that region R_l represents a larger number of hits and accommodates a larger variance in

the input space. Similarly region R_s represents a smaller number of hits and a smaller variance. Thus R_l represents a larger portion of the input space and R_s represents a smaller portion. Therefore we can infer that in the case of a low SF value, node i represents a larger region of the input space and with a high SF value, node i represents a smaller region. By generalising i to be any node in the GSOM it can be concluded that, with a small SF value, the nodes in the GSOM represents larger portions of the input space and with a high SF value, the nodes represent smaller portions. Therefore using the same input data, a low SF value will produce a smaller representative map and a high SF value will produce a larger representative map.

In the case of the SOM, the spread of the map is pre-determined by the data analyst and the input data in a certain direction is forced to fit into the available number of nodes. As such, unless the analyst has a method of (or knowledge of) assessing the *proper* number of nodes, a distorted map can occur. With the GSOM, the input values dictate the number of nodes and the SF provides global control of the spread of the nodes. Therefore the GSOM does not result in the oblique orientation or distorted view of the data as in the case of SOM.

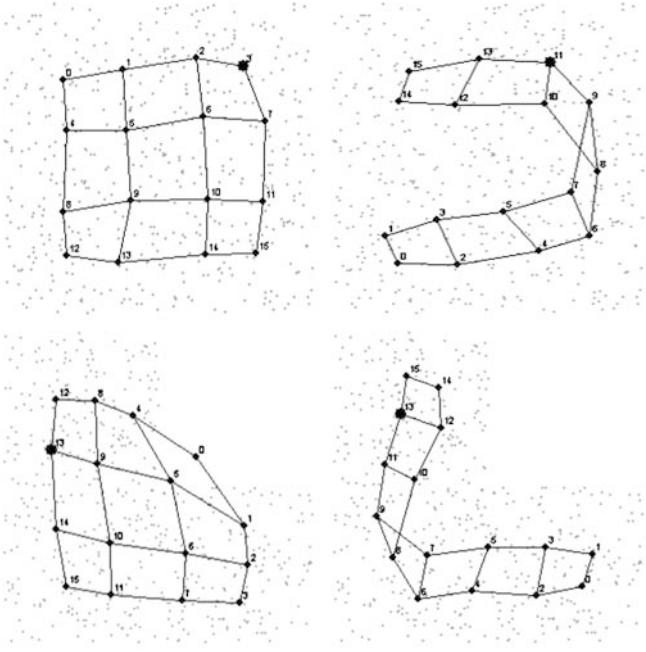

Fig. 4 Square and L-shaped data sets mapped to the SOM

4 Experiments

In this section we present several sets of experimental results to highlight the differences between GSOM with the SF and the traditional SOM. Three artificial data sets with uniform distributions of *square, L-shaped* and *star* two-dimensional data are used. The first experiment (Fig. 4) show the results from using the SOM on the two data sets. The small dots in the figures represent the input data distribution. The figures show that the shape of the grid effects the level of match to the input data distribution.

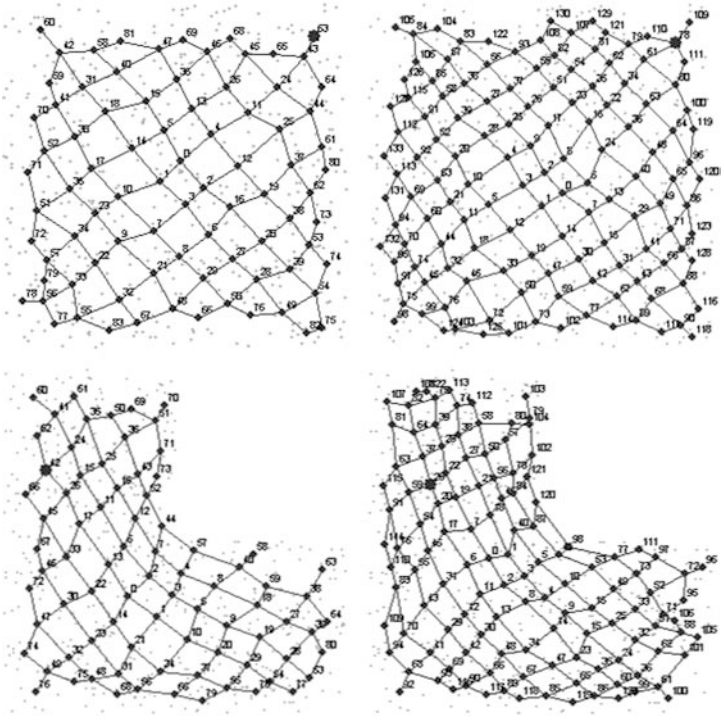

Fig. 5 Square and L-shaped data sets mapped to the GSOM

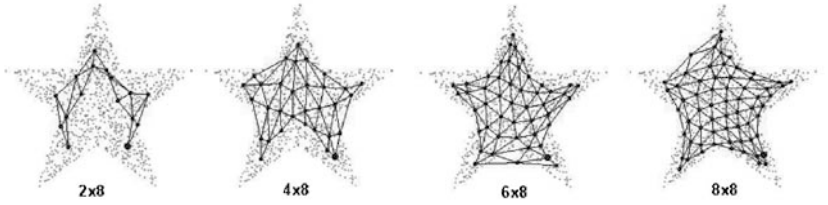

Fig. 6 Different SOM structures mapped with star data

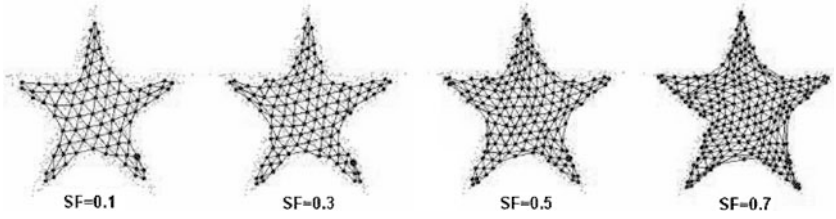

Fig. 7 GSOM with different SF mapped with star data

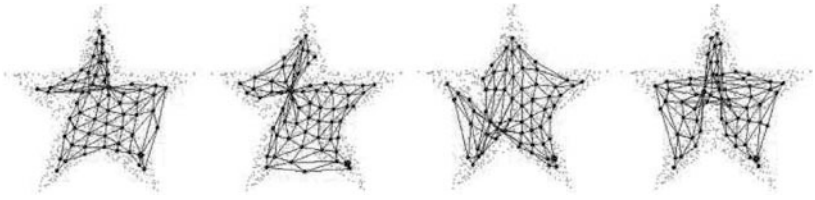

Fig. 8 Twisted SOMs

Figure 5 shows the same data sets mapped to the GSOM. Because the GSOM increases the grid size using the SF, we have shown two sets of GSOMs with 0.1 and 0.5 spread on the square and L-shaped data. These experiments show that the shape of the GSOM grid is decided by the input data distribution.

Figures 6, 7 and 8 show further experimental results demonstrating the behaviour of the GSOM, controlled with the SF, in comparison with the fixed-structure SOM.

Figure 6 shows four SOMs with different grid structures mapping a star-shaped uniform data set. It is clearly seen that as the grid structure becomes more and more *square* shaped, the networks fits the input data better (since the input star shape is closer in shape to a square than a rectangle). Figure 7 shows four GSOMs with different SF values on the same star shaped data set. It is clearly seen that the effect of the SF is to increase the number of nodes, still maintaining the same two-dimensional shape. Because the software used does not have the facility of visualising maps of larger size the higher number of nodes are shown as a more dense map, instead of a larger star shape. Another major advantage of the GSOM is reduction of map twisting; Fig. 8 demonstrates this effect. The GSOMs in figure were obtained with a single run, while the maps in figure required several runs to obtain a *proper* map without twists. Figure 8 shows several of the twisted SOMs obtained when attempting to acquire the 8×8 SOM.

5 Conclusions

In this paper we have highlighted the advantages of the parameter called the SF in the GSOM model. The paper attempts to describe the difference between the GSOM and the traditional SOM in terms of obtaining maps which fit in to the data distributions. In the traditional SOM the user has to experiment with different grid sizes to arrive at a *best fitting* grid. The *rows to columns* ratio of the grid needs to be initialised to match the data distribution, otherwise a distorted map is obtained. We show that the GSOM with the SF takes this burden of pre-defining the *correct* grid size off the user. Therefore we describe the GSOM as a better alternative in generating maps which fit the input data distribution.

Acknowledgements The author acknowledges that Figs. 4 and 5 were obtained using the GSOM program developed by Mr. Arthur Hsu, University of Melbourne, Australia.

References

1. Alahakoon LD (2000) Data mining with structure adapting neural networks. PhD thesis, School of Computer Science and Software Engineering, Monash University
2. Alahakoon LD, Halgamuge SK, Srinivasan B (1998) A self growing cluster development approach to data mining. In: Proc IEEE conf on systems man and cybernetics, pp 2901–2906

3. Alahakoon LD, Halgamuge SK, Srinivasan B (1998) A structure adapting feature map for optimal cluster representation. In: Proc int conf on neural information processing, pp 809–812
4. Alahakoon LD, Halgamuge SK, Srinivasan B (2000) Dynamic self-organising maps with controlled growth for knowledge discovery. IEEE Trans Neural Netw 11(3):601–614
5. Blackmore J, Miikkulainen R (1996) Incremental grid growing: encoding high dimensional structure into a two dimensional feature map. In: Simpson P (ed) IEEE technology update. IEEE Press
6. Fritzke B (1994) Growing cell structure: a self organising network for supervised and un-supervised learning. Neural Netw 07:1441–1460
7. Kohonen T (1995) Self-organizing maps. Springer, Berlin Heidelberg New York
8. Martinetz TM, Schultan KJ (1944) Topology representing networks. Neural netw 7(3):507–522