Comparison of dynamic feature map models for environmental monitoring *

T. Trautmann and T. Denœux

Université de Technologie de Compiègne
U.R.A CNRS 817 Heudiasyc
BP 649 - 60206 Compiègne cedex - France
Phone: (33) 44 23 44 23 Fax: (33) 44 23 44 77
email: trautman/tdenoeux@hds.univ-compiegne.fr

Abstract

The Self Organizing Feature Map (SOM) model developed by Kohonen has the ability to generate a mapping from a possibly high-dimensional feature space onto a discrete two-dimensional space of activations in a lattice of neuron-like processing elements. Although this model has proved efficient in a variety of applications, it suffers from some limitations such as lack of adaptation of the map architecture. This problem is particularly acute in on-line monitoring applications, in which the input distribution varies as a function of time, and the training set cannot be assumed to be representative of all possible system states. In this paper, three topology feature map models with architecture adaptation are applied to water quality data and compared to the basic SOM model with fixed architecture using five objective performance criteria. The BungySOM model is shown to outperform the other constructive algorithms and to carry out faster adaptation to new inputs as compared to the SOM model.

1 Introduction

The SOM model introduced by Kohonen [7] allows to define a mapping from an input space of arbitrary dimension onto a one or two-dimensional space of activations in a lattice of neuron-like processing elements. This mapping has the property of topology preservation, i.e., two neighboring input vectors are mapped onto neighboring nodes in the lattice. However, the training algorithm assumes a certain map architecture (size and shape), which has to be designed *a priori*. In the case of low-dimensional inputs, it is possible to perform subjective evaluation of trained maps using graphical displays. In higher dimensional settings, data exploration techniques such as Sammon's mapping can provide rough indications [8]. This approach however does not constitute a systematic and objective method to determine the optimal architecture or compare several maps obtained with different initial conditions and learning algorithms. There is therefore a need for objective performance criteria.

Another related problem is the lack of adaptation of the map architecture to new inputs. Such an adaptation is needed in monitoring applications in which the training set does not contain any pattern relative to certain states of the system under study. Updating of the map in response to such patterns without modification of the architecture may in some cases degrade the representation of previously learnt input vectors. This justifies the development of constructive algorithms allowing to grow new units when the input vector is very dissimilar from existing prototypes.

A first attempt to perform an objective comparison of several static and dynamic feature map training algorithms was presented by Fritzke [5] using artificial data sets. This paper extends this approach by presenting additional quality criteria and using real data as a basis for comparison. Three constructive algorithms are considered, including a new version of the BungySOM model [9]. The paper starts with a presentation of the quality criteria (Section 2), and the dynamic growing models considered in this study (Section 3). Experimental results are presented and discussed in Section 4. Section 5 concludes the paper.

*Proceedings of the *International Conference on Neural Networks (ICNN'95)*, volume 1, pages 73–78, Perth, Australia, November 1995. IEEE.

2 Performance criteria

2.1 Existing criteria

A topology preserving feature map is composed of N cells represented in the input space $V \subset I\!R^n$ by their weight vectors $\mathbf{w}_i \in I\!R^n, i = 1, \ldots, N$. Cell i is said to be "activated" for a given input $\mathbf{x} \in V$ if $d^V(\mathbf{x}, \mathbf{w}_i) = \min_{j \in \{1,\ldots,N\}} d^V(\mathbf{x}, \mathbf{w}_j)$, where $d^V(\cdot)$ denotes the Euclidean distance measure in V. It is then referred to as the *best matching unit* (*bmu*).

In the learning algorithm proposed by Kohonen, as in other vector quantization methods, weight vectors are moved according to a certain learning process, in order to reduce the quantization error. The most interesting property of the model is the organization process that takes place during learning and allows the map to model the topology of the input space V. The results of both processes have to be taken into account while assessing the quality of a given map.

For the vector quantization part, the global quantization and distribution errors are usually considered, as proposed by Fritzke [5]. For the organization process, we have to check if the map preserves the input space topology and if the map is well organized. Let us denote by Θ_A^V the mapping from input space V to the output space A performed by the map. As a neighborhood is defined for each cell, the learning process [7] allows the map to reach an organization that preserves the topology of the input space. The preservation of topology by Θ_A^V means that two close input samples must be represented on the map by two neighboring cells. The *topographical product* [1] (topo p) is used in this paper to measure the quality of topology preservation. Additionally, one also has to check that the map is well organized: two neighboring cells in A must have close weight vectors in V (${\Theta_A^V}^{-1}$ is then said to preserve the neighborhood). The D_{Map} criterion was introduced by Zrehen and Blayo [10] for that purpose. Other criteria have been proposed such as the mean edge length [5] or visual examination of graphical displays [3]. In the following, we propose a new criterion based explicitly on a rigorous definition of organization.

2.2 New criterion

Let us define the rank k neighborhood of cell i by $N_k^A(i) = \{j \in A | d^A(i,j) \leq k\}$, where $d^A(i,j)$ is defined as the smallest number of neighborhood links between cells i and j. The rank k neighborhood of cell i is said to be *preserved* if:

$$\forall h \notin N_k^A(i), d^V(\mathbf{w}_h, \mathbf{w}_i) > \max_{j \in N_k^A(i)} d^V(\mathbf{w}_j, \mathbf{w}_i) \quad (1)$$

A vector $\mathbf{y} \in V$ *disturbs* the rank k neighborhood of cell i if, for some $j \in N_k^A(i)$, $d^V(\mathbf{y}, \mathbf{w}_i) \leq d^V(\mathbf{w}_j, \mathbf{w}_i)$.

A map is said to be *well organized* if all cells have their rank k neighborhood preserved, for any value of k. Equation 1 suggests a local criterion R_i^k defining the number of times a given rank k neighborhood of a cell i is not preserved:

$$R_i^k = \text{card}(\{h \notin N_k^A(i), \exists j \in N_k^A(i), d^V(\mathbf{w}_h, \mathbf{w}_i) \leq d^V(\mathbf{w}_j, \mathbf{w}_i)\}) \quad (2)$$

If we now consider all cells and neighborhoods ranks, we obtain a global criterion:

$$R = \sum_{i=1}^{N} \sum_{k=1}^{k_{max}} R_i^k \quad (3)$$

where k_{max} denotes the maximum neighborhood rank for the given architecture. If $R = 0$, the map is perfectly organized. Otherwise, there is a cell whose rank k neighborhood is not preserved for some value of k. To reduce calculation time, we can use the local criterion R^1 that only considers the direct neighborhood ($k = 1$) of each cell: $R^1 = \sum_{i=1}^{N} R_i^1$. The normalized values $R/(N * k_{max})$ and R^1/N should be used to compare several maps with different architectures.

3 Dynamic growing models

3.1 Existing models

When a sufficiently "representative" learning set is available, one can define a SOM architecture, train it and apply it to new incoming data. As argued by Kohonen [6], a map with arbitrary architecture allows to extract relevant information from a given data set, so that there is no need for architecture adaptation in that case. In some real applications however, the training set contains only partial information concerning

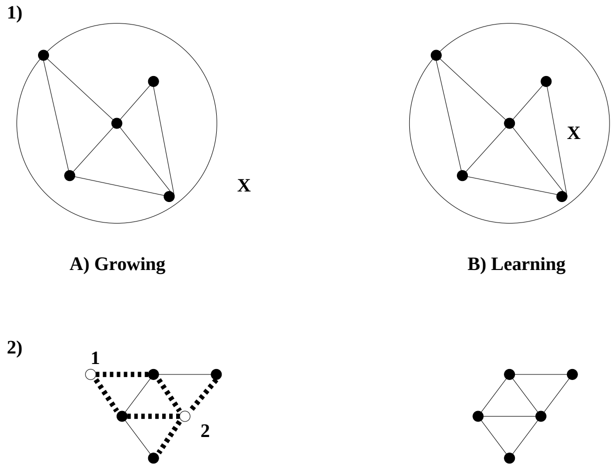

Figure 1: 1) Responses to a new input vector in two different situations: creation of a new unit (left) and updating of existing weight vectors (right). 2) Placement heuristic for insertion of a new cell in the grid position 2 is prefered over position 1 because it creates more neighborhood links.

possible inputs. For example, in diagnosis applications, the initial data set may not contain measurements relative to certain system states. When new data samples become available, it is then necessary to re-train the map with an updated learning set in order to maintain the accuracy of the model. However, if one does not adapt the map architecture, this accuracy (quantization and distribution error) usually decreases. In response to this problem, dynamic models with adaptive architectures have been proposed by Fritzke [5] (Growing Cell Structure - GCS) and Blackmore [2] (Incremental Grid Growing - IGG). These models grow architectures depending on the structure of the learning set, based on user-defined parameters such as required accuracy and final number of cells in GCS, and connect and disconnect threshold parameters in IGG. Both are working in *batch* mode: a new cell is added after a given amount of input samples have been presented to the network. Consequently, they cannot easily be used in monitoring applications, in which one does not know in advance how many cells will be necessary to map unknown process states not yet represented in the data. We therefore developed a new *on-line* dynamic growing model [9]. The latest version of this model is presented in the next section.

3.2 BungySOM

BungySOM is an on-line model of self-organizing map construction and adaptation. The learning process starts with a unique cell whose weight vector is initialized with the first input vector. The map organization criterion R^1_{bmu} is used to decide whether a new cell should be added to the map (Figure 1). As soon as a cell becomes bmu, its activation frequancy $\tau_{bmu}(t)$ is incremented; this value is decremented by a rate $\delta, 0 < \delta < 1$ for all the other cells [4]:

$$\tau_{bmu}(t+1) = \tau_{bmu}(t) + 1 \tag{4}$$
$$\tau_i(t+1) = \tau_i(t) - \delta.\tau_i(t), \forall i \neq bmu \tag{5}$$

For each input vector $\mathbf{x} \in V$, a new cell new is grown if the following conditions are verified:

1. the direct neighborhood of bmu is preserved: $d^V(\mathbf{x}, \mathbf{w}_i) > \eta \max_{j \in N_1^A(i)} d^V(\mathbf{w}_j, \mathbf{w}_i)$, where η is a parameter controlling the growing speed and accuracy of the map;
2. the bmu belongs to the map border (it has, at least, one free neighboring position on the grid);
3. the bmu activation frequency is higher that the average one over the map.

In all other cases, the weight vectors of the best matching unit and its neighbors are simply updated. A new cell activation frequency is initialized with: $\tau_{new}(t) = 1$.

The BungySOM model is defined to work on diagnosis applications with continuous arrival of new input vectors. In order to maintain some plasticity of the map even after a large number of input presentations, it is no longer possible to define the *learning factor* $\alpha(t)$ and the *neighborhood radius* $h_{bmu,i}(t)$ as functions of time, as in the classical SOM learning rule [7]. For this reason, we defined both as functions of cell activation frequency τ_i. The learning rule becomes:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha_{bmu,i}(t) \cdot h_{bmu,i}(t) \cdot (\mathbf{x} - \mathbf{w}_i(t)) \tag{6}$$

for all $i \in \{1, \ldots, N\}$. The neighborhood radius $h_{bmu,i}$ is defined for $d^A(bmu, i) \leq 2$ as:

$$h_{bmu,i}(t) = \exp(-\frac{d^A(bmu,i)}{2\sigma^2}) \tag{7}$$

with $\sigma = \sigma_0/\tau_{bmu}^{0.1}$. The learning factor $\alpha_{bmu,i}(t)$ is a function of the activation frequances of units i and bmu:

$$\alpha_{bmu,i}(t) = \frac{\alpha_0}{\tau_{bmu}^{0.2} \cdot \tau_i^{0.2}} \text{ if } i \neq bmu \tag{8}$$

$$\alpha_{bmu,i}(t) = \frac{\alpha_0}{\tau_{bmu}^{0.2} \cdot \tau_{bmu}^{0.2}} \text{ if } i = bmu \tag{9}$$

To allow easy interpretation of the map, the maximum number of direct neighbors is defined as 6 (hexagonal neighborhood) or 4 (rectangular neighborhood), as in the SOM and IGG models. In "batch" mode, BungySOM stops when no cell creation has occurred after a specified number of training set presentations. In "on-line" mode, BungySOM continuously adds new cells each time the conditions for cell creation are met. The new cell is placed in A in the direct neighborhood of the bmu, so as to create a maximum number of connections (Figure 1). The weight vector $\mathbf{w}_{new}$ is defined as:

$$\mathbf{w}_{new} = \frac{1}{n_1+1}\left(\mathbf{x} + \sum_{i \in N_1^A(new)} \mathbf{w}_i\right) \tag{10}$$

with $n_1 = \text{card}(N_1^A(new))$. To obtain a better representation of data clusters, the possibility is left to prune unused cells and their neighborhood links. A cell i is considered as *obsolete* if its activation frequency τ_i is lower than a given threshold ν. It is also possible to reconnect cells: if the direct neighborhood of cell i is disturbed by a cell j that is not connected to i but placed on a direct neighboring grid location, then i and j are connected.

The algorithm can be summarized as follows:

1. WHILE not (stop criterion) DO:
   (a) Present an input sample $\mathbf{x}$.
   (b) Select the Best Matching Unit bmu: $d^V(\mathbf{x}, \mathbf{w}_{bmu}) = \min_{i \in \{1,\ldots,N\}}(d^V(\mathbf{x}, \mathbf{w}_i))$.
   (c) Update the activation frequences:
   $$\tau_{bmu}(t) \leftarrow \tau_{bmu}(t) + 1$$
   $$\tau_i(t) \leftarrow \tau_i(t) - \delta.\tau_i(t), \forall i \neq bmu$$
   (d) IF $d^V(\mathbf{x}, \mathbf{w}_i) > \eta \max_{j \in N_1^A(i)} d^V(\mathbf{w}_j, \mathbf{w}_i)$ AND bmu belongs to the map border AND $\tau_b(t) >$ mean($\tau_i(t)$)) THEN grow a new cell new, $\tau_{new} \leftarrow 1$ ELSE apply the modified learning rule (eq.6).
2. Prune obsolete cells: IF cell i has $\tau_i < \nu$ THEN prune it and the links to its direct neighbors.
3. $\forall i, j \in \{1, \ldots, N\}$ not connected together: IF j disturbs the direct neighborhood of i AND $j \in N_1^A(i)$ then connect i and j.
4. END WHILE.

4 Experiments

The data set used in this experiment consists of daily measurements of four water quality parameters (temperature, pH, NH_4 and NO_2 concentrations) performed in the river Seine during two years. The water

Table 1: Performance criteria (means and standard deviations) for four different feature map models on water quality data.

Model		**Criteria**				
	map size	Q error	Dist err	Topo p	DMAP	R1
SOM	8 x 9	0.53	0.79	0.009	0.25	2.26
SOM	7 x 7	0.68	0.76	0.027	**0.20**	**1.40**
GCS	50 cells	**0.26**	0.58	0.09	0.61	8.57
std dev	*0*	*1.e-4*	*9.e-4*	*5.e-5*	*0.09*	*2.63*
IGG	50 cells	0.34	**0.54**	0.031	0.45	4.47
std dev	*0*	*4.e-7*	*8.e-4*	*2.e-5*	*0.01*	*0.87*
BungySOM	53 cells	0.31	0.55	**0.003**	0.23	3.53
std dev	*72*	*1.3.e-3*	*5.e-3*	*7.e-5*	*6.e-4*	*0.64*

Table 2: Performance criteria for the SOM and BungySOM models on the water quality data with sequential presentation of the data in time order. The results are presented after 1, 5 and 10 presentations of the learning set, respectively.

Model		**Criteria**				
	map size	Q error	Dist err	Topo p	DMAP	R1
SOM 1	8 x 9	2.03	1.36	0.028	1.21	7.87
BungySOM 1	50 cells	**0.6**	**0.9**	**0.003**	**0.7**	**6.2**
SOM 5	8 x 9	1.03	1.32	0.009	**0.39**	**2.34**
BungySOM 5	74 cells	**0.38**	**0.82**	**0.003**	0.4	4.43
SOM 10	8 x 9	1.10	1.36	0.015	0.62	6.05
BungySOM 10	86 cells	**0.31**	**0.69**	**0.001**	**0.29**	**4.12**

quality monitoring system to be developed will operate in real time near Paris with on-line measurements of the same parameters.

This data set was used as a basis for comparison of the three models described above (IGG, GCS, BungySOM) and the standard SOM model with fixed architecture. The parameters of the BungySOM model were set to $\eta = 1.15, \delta = 0.005$ and $\nu = 0.05$, yielding an average of 53 cells (with standard deviation of 72) in 5 runs with different initial conditions. The two other dynamic models were tuned to grow 50 cells. For the GCS model we adopted the parameter values recommended by Fritzke [5]. The two architectures tested for the SOM model were determined empirically using subjective interpretation of Sammon projections.

Table 1 shows the mean and standard deviations of the five performance criteria mentioned in Section 2. For each of these criteria, a value close to zero indicates good performance. The four training algorithms were iterated until convergence. These results reveal a good overall performance of the maps generated by the dynamic models, which are only outperformed by the SOM according to the map organization criteria (DMAP, R1). The BungySOM map provides the best results of all three dynamic models according to three criteria, and comes in second position according to the quantization and distribution error.

In a second experiment, the training data were presented sequentially *in time order* to the SOM and BungySOM algorithms with $\eta = 1, \delta = 0.005$ and $\nu = 0.05$. The performance criteria were computed after 1, 5 and 10 presentation of the 731 training patterns, respectively (Table 2). The BungySOM model was found to adapt much faster than the SOM model to new inputs, resulting in lower values of all five performance criteria after a small number of learning cycles. This can be regarded as a substantial advantage of the method in on-line monitoring applications, in which the network has to quickly adapt to new input patterns corresponding to system states which have not been encountered before.

5 Conclusion

The ability of a topology preserving feature map to quickly adapt its architecture to new inputs is regarded as essential in process monitoring and fault diagnosis applications. This justifies the choice of dynamic models which are able to grow new cells in response to atypical patterns, thereby realizing a good compromise between stability and plasticity. Three different dynamic models have been tested on real data set and compared to the basic SOM model with fixed architecture, using five objective performance criteria. Although the constructive algorithm did not allow to attain quite the same level of performance as the SOM model in terms of map organization and topology preservation, the BungySOM model yielded satisfactory results according to the five quality criteria and globally outperformed the two other dynamic models. Moreover, this algorithm was shown to allow faster adaptation to new inputs, as compared to the SOM model. Further work will concentrate on the validation of the BungySOM model using on-line measurements collected every 15 min. The next step will be the implementation of the model for operational purposes.

Acknowledgment

This work was partially supported by EEC funded Esprit Project 6757 EMS (Environmental Monitoring System).

References

[1] H. U. Bauer and K. Pawelzik. Quantifying the neighborhood preservation of the self organizing feature map. *IEEE trans on Neural Networks*, 3:570–79, 1992.

[2] J. Blackmore and R. Miikulainen. Incremental grid growing: encoding high dimensional structures into a two dimensional feature map. Technical Report TRAI92-192, University of Texas at Austin, 1992.

[3] P. Demartines. Mesures d'organisation du réseau de Kohonen. In *Congrès Satellite du congrès Européen de Mathématiques*, Paris, 1992.

[4] B. Fritzke. Growing cell structures a self organizing network for unsupervised and supervised learning. Technical Report TR-93-026, ICSI, 1993.

[5] B. Fritzke. Kohonen feature maps and growing cell structures - a performance comparison. In S. J. Harson C. L. Giles and J. D. Cowan, editors, *Advances in neural information processing systems 5*, pages 123–30. Kaufmann San mateo CA, 1993.

[6] T. Kohonen. What generalization of the SOM make sense. In M. Marinaro and G. Morasso, editors, *Proceedings of ICANN 94 Soronto Italy*, pages 292–97. Springer Verlag.

[7] T. Kohonen. *Self Organizing Maps*, volume 30 of *Springer series in information sciences*. Springer Verlag, 1995.

[8] J. Kangas T. Kohonen, J. Hynninen and J. Laaksonen. *The Self Organizing Map program pakage SOM PAK V3.0*. SOM programing team - Helsinki Univ of Tech - Lab of computers and Info Science, 1995.

[9] T. Trautmann and T. Denœux. A constructive algorithm for S. O. M applied to water quality monitoring. In C. H. Dagli, B. R. Fernandez, J. Gosh, and R. T. Soudar Kumara, editors, *Intelligent Engineering systems through Artificial Neural Networks*, volume 4, pages 17–22. ASME PRESS, 1994.

[10] S. Zrehen and F. Blayo. A geometric organization measure for Kohonen's map. In *Proceedings of Neuro Nimes*, pages 603–10, November 1992.