Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, February 15-17, 2006 (pp18-22)

A Trajectory Index For Topology Conserving Mapping

HILARIO LÓPEZ and IVÁN MACHÓN and EVA FERNÁNDEZ
Departamento de Ingeniería Eléctrica, Electrónica de Computadores y Sistemas
Universidad de Oviedo
Edificio Departamental 2. Zona Oeste. Campus de Viesques s/n. 33204 Gijón (Asturias)
SPAIN
http://isa.uniovi.es/~hilario

Abstract: - Topology preservation during training of Self-Organizing maps is a key factor to obtain a good quality visualization in process monitoring. In this paper, different existing indexes for topology preservation measurement are exposed and also a proposed new index is formulated.

Key-Words: - Self-organizing mapping, validation, topological property, process monitoring, trajectories, input space.

1 Introduction

Self-Organizing Map (SOM) is usually used to develop a model for process monitoring. The reliability of this monitoring task, in part, depends on the topology preservation of the data set during the training. If this property were not respected, the projection of the process state in the map would follow random and chaotic trajectories, disabling the process monitoring.

The main objective of this paper is to present a review of the principal existing indexes for topology preservation measurement and to formulate a proposed new index based on the idea of the smoothing trajectory occurred under non-abruptly process state variation.

2 SOM

The SOM [1] consists of a regular lattice typically defined in a two dimensional space composed of several neurons placed in the nodes of the lattice. SOM training implies assigning a set of coordinates in the input data space, which are called codebook vectors, to each neuron. Thus, each neuron is represented by a codebook vector and a correspondence is established between the coordinates of each neuron in the input space (data set) and their coordinates in the 2D-lattice or output space. This study was carried out using a batch training algorithm for SOM training. This algorithm is iterative and does not use any learning rate, but instead of using only one data vector at a time, the whole data set is used before updating the codebook vectors (1). For this reason, the algorithm is called batch.

$$m_i(t+1) = \frac{\sum_{j=1}^{N} h_{bi}(t) \cdot x_j}{\sum_{j=1}^{N} h_{bi}(t)} \quad (1)$$

$$h_{bi}(t) = \mathrm{e}^{-\frac{\|r_b - r_i\|^2}{2\cdot\sigma^2(t)}} \quad (2)$$

where b is the best matching neuron of data vector x_j and $h_{bi}(t)$ is the neighborhood function value of the data vector i with the kernel centered at the winner unit b. The updated codebook vector is a weighted average of the data vectors. In equation (2) $h_{bi}(t)$ is a neighborhood kernel centered in the winner neuron b and it is usually assumed as Gaussian where r_b and r_i are positions of neurons b and i on the SOM grid or output space and $\sigma(t)$ is a neighborhood radius or also the standard deviation of the Gaussian and decrease monotonically with time.

Moreover, the codebook vectors approximate to the data set trying to substitute a data vector for a codebook vector of the SOM. A consequence of this approach is the quantization error. Equation (3) is usually used to calculate the average quantization error over the whole data set. N is the number of samples, x_i is the ith data sample and m_b is the codebook vector of the best matching neuron for x_i.

$$e_q = \frac{1}{N} \sum_{i=1}^{N} \|x_i - m_b\| \quad (3)$$

The work developed in this paper uses equations (4) and (5) to determine the output space size [2][3]. The number of neurons of the output space is determined by equation (4). M is the number of neurons and N is the number of samples of the training data.

$$M = 5 \cdot \sqrt{N} \tag{4}$$

On the other hand, the criterion of the utilized toolbox to determine the ratio between the number of rows n_1 and the number of columns n_2 of the 2D grid or output space is calculated according to equation (5). The ratio between sidelengths of the map is the square root of the ratio between the two biggest eigenvalues of the training data. The highest eigenvalue is e_1 and the second highest is e_2.

$$\frac{n_1}{n_2} = \sqrt{\frac{e_1}{e_2}} \tag{5}$$

3 Topographic errors

According to the properties of the SOM, the trained neural network must achieve the topology preservation of the data. Therefore the neighborhood on the output space and in the input space must be similar. If two codebook vectors close to each other in the input space are mapped wide apart on the grid, this is signaled by the situation where two closest best matching neurons of an input vector are not adjacent neurons. This kind of fold is considered as an indication of the topographic error in the mapping and does not verify the SOM property about training data topology preservation where neighbor neurons of the output space correspond to similar values of the process variables, i. e., regions of the output space represent working zones of the process.

Different existing indexes for topology preservation measurement are presented bellow in the following subsections.

3.1 Kiviluoto error

The topographic error [4] can be calculated by equation (6) as the proportion of sample vectors for which two best matching neurons are not adjacent. N is the number of samples, x_k is the kth sample of the data set and $u(x_k)$ is equal to 1 if the first and second best matching neurons of x_k are not adjacent neurons, otherwise zero.

$$e_t = \frac{1}{N} \sum_{k=1}^{N} u(x_k) \tag{6}$$

The results of this error measurement are very easy to interpret and are also directly comparable between different models and even mapping of different data sets. The main drawback is that it does not take into account the metric relations with non-neighbors.

3.2 Topographic Product

A classical measure for topographic preservation is the topographic product introduced in [5]. For each neuron j the sequences $n_k^{OS}(j)$ and $n_k^{IS}(j)$ must be determined, where $n_k^{OS}(j)$ denotes the k*th* neighbor neuron of neuron j, which has been quantified using Euclidean distance between the neurons measured in the output space; whereas, $n_k^{IS}(j)$ denotes the neuron corresponding to the k*th* neighbor codebook vector of codebook vector j, which has been quantified using Euclidean distance between the codebook vectors, w_j and $wn_k^{IS}(j)$, measured in the input space. The final result is calculated over all the sequences for each neuron j according equation (10), where N is the number of neurons in the map.

$$Q_1(j,k) = \prod_{l=1}^{k} \frac{D^{IS}(w_j, wn_k^{OS}(j))}{D^{IS}(w_j, wn_k^{IS}(j))} \tag{7}$$

$$Q_2(j,k) = \prod_{l=1}^{k} \frac{D^{OS}(j, n_k^{OS}(j))}{D^{OS}(j, n_k^{IS}(j))} \tag{8}$$

$$Q(j,k) = \left(\prod_{l=1}^{k} Q_1(j,k) \cdot Q_2(j,k) \right)^{\frac{1}{2k}} \tag{9}$$

$$P = \frac{1}{N(N-1)} \sum_j \sum_k log\big(Q(j,k)\big) \tag{10}$$

3.3 Bezdek error

This coefficient gives an estimation of topology preservation based on a correspondence of the sequences or rankings established by all distance pairs that take place in the input and output space [6][7]. The description of the algorithm emphasizes continuity and isometry as properties of the coefficient. The continuity preserves neighborhoods but not distance order. However, isometry preserves also distances.

A metric topology preserving transformation is achieved if and only if for any codebook vector in the input space w_i, whenever w_j is the k*th* nearest neighbor of w_i (using the metric defined previously), then neuron j is the k*th* nearest neighbor of i in the output space, taking into account the metric defined of this one.

A matrix of distances D^{IS} between any two codebook vectors of the input space must be formulated to carry out the proposed algorithm. Then a vector d^{IS} is obtained from matrix D^{IS} representing the same distances. In the same way the matrix of distances

between neurons D^{OS} in the output space and its vector d^{OS} are obtained. Both vectors, d^{OS} and d^{IS}, have length equal to T according equation (12) where N is the number of neurons in the map. Two ranking vectors, r^{OS} and r^{IS}, and , are obtained from vectors d^{OS} and d^{IS}, respectively by replacing the components by their ranks in the sequence of distances. For example, a value equal to 1 means the smallest distance.

Bezdek and Pal's coefficient of topology preservation is quantified by a statistical measure for the degree of correlation between ranking orders using Spearman's ρ according equation (11) [8].

$$\rho(r^{OS}, r^{IS}) = 1 - \frac{6\sum_{k=1}^{T}\left(r^{OS}(k) - r^{IS}(k)\right)^2}{T^3 - T} \qquad (11)$$

$$T = \frac{n(n-1)}{2} \qquad (12)$$

If there are cases of ties in the rank, tied ranks can be replaced by their average, see [8].
Perfect ordering turns out d^{OS} equal to d^{IS}, and therefore ρ=1. Completely random ordering yields ρ<1 and a reversed ordering ρ=-1.

3.4 The Zrehen-measure

This index measures the topology preservation in the following way: A pair of neighbor neurons i and j is locally organized if the straight line joining their weight vectors w_i and w_j contains points which are closer either to w_i or to w_j than they are to any other [9].

From a geometric point of view, a sphere with radius $|(w_i - w_j)/2|$ and center $|w_i + w_j|/2$ must not contain any other codebook vectors. The codebook vectors w_k violating equation (13) are called "intruders", and the Zrehen measure is the sum of all intruders over all pairs of neighboring neurons.

$$\|w_i - w_k\|^2 + \|w_k - w_j\|^2 \le \|w_i - w_j\|^2 \qquad \forall k \ne i, j \qquad (13)$$

In [9] a normalization procedure to accommodate different numbers of neurons, and therefore different map sizes, was not described. In this work the overall sum of intruders is divided by the number of neuron pairs.

4 Trajectory index proposed

Interesting ideas can be extracted from the property about topology preservation of the data. The neighborhood relations between codebooks vectors in the input space and lattice neurons in the output space must be similar.

When projecting the current values of the process variables, assigning the best matching neuron to each of them, this topology preservation property implies an important relationship. In this way, the projection of the process state is obtained. Based on the topology preservation property, similar values of the process variables correspond to best matching neurons that are neighbor each other. Therefore, while the process state does not vary abruptly, its projection also will not change abruptly in the map. Then, the regions of the output space can be identified as different process zones from a operating point of view. In fact, this important property makes SOM algorithm a powerful tool for supervision of multivariable processes [10][11][12]. If this property were not respected, the projection of the process state in the map would follow random and chaotic trajectories, disabling the process monitoring.

The proposed index that is presented in this work is developed taking into account this geometric interpretation of the trajectories of the process state projection on the output space. An optimal map will carry out smooth trajectories under small changes of one process variable.

The index is calculated by means of a trajectory on the output space obtained from the best matching neurons corresponding to a variation of a process variable overall its range keeping constant the rest of variables of the data set. The number of nodes along the trajectory, it is to say, the number of best matching neurons N is calculated. The higher the number of nodes the better resolution of the map. The number of directions (north, south, west, east, northwest, northeast, southwest and southeast) that are taken along the trajectory from node to node are considered. If opposite directions are taken, the index must be incremented. Also the higher the number of directions taken along the trajectory, the higher the index.

$$Trajectory_{index} = \sum_{variables}\left(\frac{1}{N} + Nc + No\right) \qquad (14)$$

where No is the minimum of movements in opposite direction (north vs. south, east vs. west, southwest vs.

northeast and southeast vs. northwest). *Nc* is the number of directions taken along the trajectory. It will be a value between 1 and 6. In both of them, *No* and *Nc*, the intention is to minimize them, whereas the number of best matching neurons *N* taken along the trajectory must be maximized to improve the map resolution. The index is calculated over each trajectory obtained from varying each process variable from the minimum value to the maximum value of its range.

A data set was formed randomly using 400 samples of 4 variables. The fourth variable is a linear combination of the first and second one. Different maps were obtained using a batch algorithm. The map size were (13 x 8) using equations (4) and (5). The weights were initialized randomly. Twenty maps were obtained. Fig. 1 shows the index values of topology preservation for different measurements exposed above.

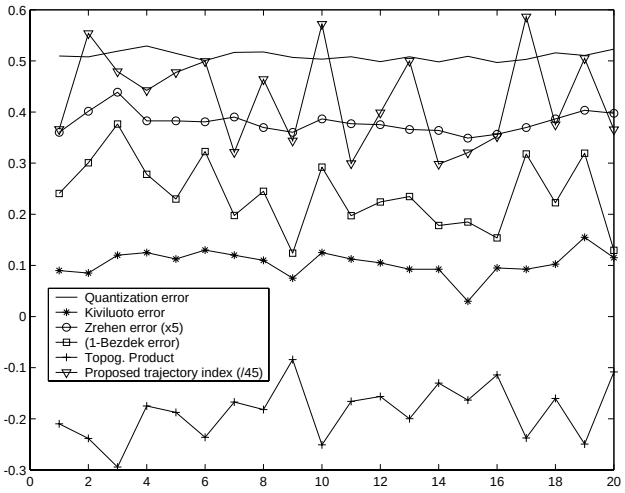

Fig. 1. Results of several indexes

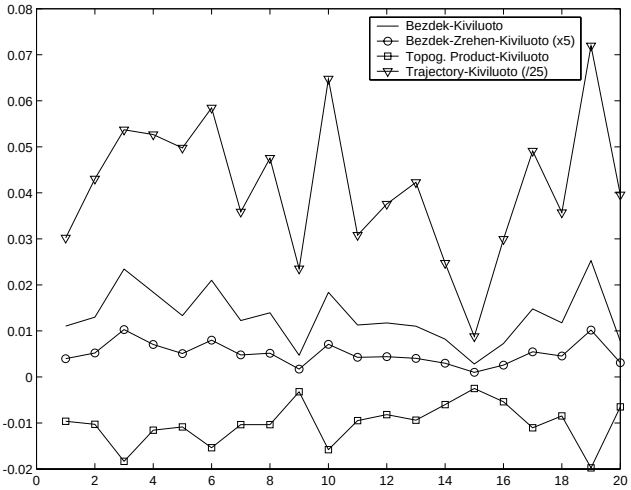

Fig. 2. Results multiplying by Kiviluoto error

Some previous works [13], based on Monte-Carlo simulations, show a high insensitivity of the SOM to the choice of initial values. The results of the quantization error seem to be independent of the initial values taken by the weights. However, the topographic error depends on the initialization.

Fig 2 shows the result of multiplying each index by Kiviluoto index. The best topology preservation correspond to a map number equal to 15. The trajectories obtained for trajectory index computation are shown in Fig. 3. Smooth trajectories are obtained compared to other maps.

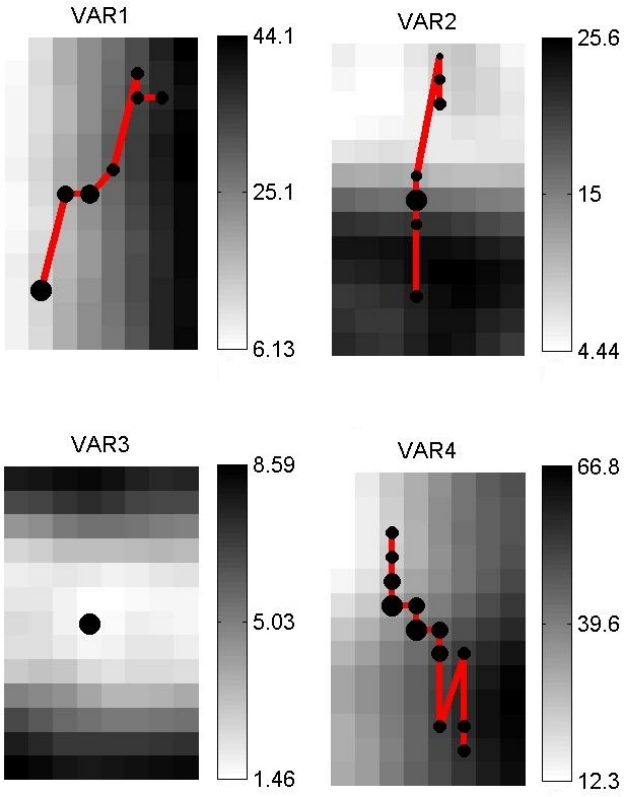

Fig. 3. Trajectories obtained for trajectory index computation

4 Conclusion

An index for topology preservation using SOM algorithm was proposed in this work. This index can be used to select the best map preserving the topology of the input data. In this way the validation of the model is carried out and the process monitoring can be improved. Reasonable results were obtained and the computational cost is not high. The results were compared to other indexes. Although most of them have a very good approach, they have high computational costs as main drawback.

References:

[1] Kohonen T. 2001. *Self-Organizing Maps*. New York: Springer-Verlag.

[2] Vesanto J.; E. Alhoniemi; J. Himberg; K. Kiviluoto and J. Parviainen. 1999. "Self-organizing map for data mining in matlab: the som toolbox," *Simulation News Europe*, pp. 25–54.

[3] López H. and I. Machón. 2004. "Self-organizing map and clustering for wastewater treatment monitoring," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 3, pp. 215–225.

[4] Kiviluoto K. 1996. "Topology preservation in self-organizing maps," in *IEEE International Conference on Neural Networks*, vol. 1, pp. 294–299.

[5] Bauer, H.-U., and Pawelzik, K. 1992. Quantifying the neighbourhood preservation of self-organizing feature maps. IEEE Transactions on Neural Networks, vol 3 (4), 570–579.

[6] Bezdek, J. C., and Pal, N. R. 1993. An index of topological preservation and its application to self-organizing feature maps. *Proc. of the IJCNN 1993, International Joint Conference on Neural Networks*, vol 3. Piscataway, NJ: IEEE Service Center, pp. 2435–2440.

[7] Bezdek, J. C., and Pal, N. R. 1995. An index of topology preservation for feature extraction. *Pattern Recognition*, vol. 28, pp. 381–391.

[8] Kendall M. and J. D. Gibbons, 1990. Rank Coorelation Methods, *Oxford University Press,* New York .

[9] Zrehen, S. 1993. Analyzing Kohonen maps with geometry. In St. Gielen & B. Kappen, *Proceedings of the International Conference on Artificial Neural Networks*, London: Springer.

[10] Machón I. and H. López. 2004. "An application for on-line control of a sequencing batch reactor," in *Proc. IFAC Workshop on Modelling and Control for Participatory Planning and Managing Water Systems*, Venice.

[11] López H. and I. Machón. 2004. "An introduction to biological wastewater treatment explained by som and clustering algorithms," in *Proc. IEEE International Symposium on Industrial Electronics*, Ajaccio.

[12] Machón I., H. López H. and A. Robles. 2005. "Treatment Stage Estimation in a Sequencing Batch Reactor" *WSEAS Transactions on Computers,* vol 4, no. 8.

[13] Cottrell, M., de Bodt, E., and Verleysen,M. 2001. "A statistical tool to assess the reliability of self-organizing maps". In N. Allinson, H. Yin, L. Allinson, & J. Slack (Eds.), *Advances in self-organising maps,* pp. 7–14. Berlin: Springer.