THE ROLE OF THE LATTICE DIMENSIONALITY IN THE SELF-ORGANIZING MAP

A.D. Ramos[*], *E. López-Rubio*[†], *E.J. Palomo*[†‡]

Abstract: The Self-Organizing Map model considers the possibility of 1D and 3D map topologies. However, 2D maps are by far the most used in practice. Moreover, there is a lack of a theory which studies the relative merits of 1D, 2D and 3D maps. In this paper a theory of this kind is developed, which can be used to assess which topologies are better suited for vector quantization. In addition to this, a broad set of experiments is presented which includes unsupervised clustering with machine learning datasets and color image segmentation. Statistical significance tests show that the 1D maps perform significantly better in many cases, which agrees with the theoretical study. This opens the way for other applications of the less popular variants of the self-organizing map.

Key words: *self-organizing map topologies, 1D map, 3D map, clustering, image segmentation*

Received: March 19, 2016 **DOI:** 10.14311/NNW.2018.28.004
Revised and accepted: February 19, 2018

1. Introduction

Since the self-organizing map (SOM) was proposed [39, 40], many variations of the original model have been developed [41]. It has always been recognized that the topology can be one, two or three dimensional, but the two dimensional version has been by far the most used in practice. This may be caused by the fact that a 2D map can be displayed on an output device directly, which facilitates the interpretation of the trained map. Visualization of high dimensional data is one of the key features of SOMs, so practitioners find easier to use 2D lattices even in applications which are not directly related to visualization because they can assess the quality of the trained maps easily. In contrast, the possibility of using 1D and 3D map lattices has received very little attention. In particular, three dimensional

[*]Antonio Díaz Ramos; Department of Algebra, Geometry and Topology, University of Málaga, Bulevar Louis Pasteur, 33. 29071 Málaga, Spain, E-mail: adiazramos@uma.es

[†]Ezequiel López-Rubio, Esteban J. Palomo – Corresponding author; Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35. 29071 Málaga. Spain, E-mail: ezeqlr@lcc.uma.es ejpalomo@lcc.uma.es

[‡]Esteban J. Palomo; School of Mathematical Sciences and Information Technology, University of Yachay Tech. Hacienda San José s/n, Urcuquí, Ecuador, E-mail: epalomo@yachaytech.edu.ec

SOMs are confined to applications where 3D data must visualized [35] or processed [12, 17, 21, 42]. However, some researchers have realized their suitability for other tasks where 3D data are not involved [7, 11, 26].

The situation for 1D SOMs is different, since in this case they are not suitable for visualization applications. The usage of these topologies is frequently linked to specific datasets where the data to be learned is known to lie in a curve, i.e. a one-dimensional manifold [2, 10, 29, 45]. But there are also some cases where the 1D SOMs are employed for general datasets where this constraint does not hold [16, 47].

A comparative theoretical study of the three above mentioned variants of the original SOM according to the lattice dimensionality has not been developed to date. Formal results about the 1D SOM are comparatively abundant [22, 23, 28, 31], because it is easier to analyze than the typical 2D SOM. Nevertheless, these works do not offer clues about which lattice dimensionality should be chosen for a particular application.

From the preceding it can be concluded that the differences among 1D, 2D and 3D lattices have not been researched adequately. It might be the case that 1D or 3D SOMs are better than 2D SOMs for applications where the visualization over a 2D output device is not necessary. Our aim here is to examine this possibility in detail, both from the theoretical and applied points of view, to assess which lattice dimensionalities are the most suitable in terms of vector quantization and topographic quality.

The structure of this paper is as follows. First the self-organizing map model is reviewed, with special attention to 1D and 3D maps (Section 2). Then the role of the lattice dimension is examined (Section 3). Experiments are shown in Section 4. Some key findings are discussed in Section 5. Finally, Section 6 is devoted to conclusions.

2. Basic concepts

In this section the fundamental concepts which this work is based on are reviewed. A brief outline of Kohonen's Self-Organizing Map [39] is presented. First the network architecture and the learning rule for the SOM are considered (Subsection 2.1). Then the energy function associated to a SOM is discussed (Subsection 2.2).

2.1 Architecture and learning rule

Here the original Kohonen's SOM is reviewed, and the notation that will be used through the paper is presented. Let M be the number of neurons of the self-organizing map, which are arranged in a lattice of size $a \times b \times c$, where $M = abc$. A 1D map lattice is obtained if we set $a > 1,\ b = c = 1$, while 2D maps correspond to $a, b > 1,\ c = 1$. Finally, 3D maps have $a, b, c > 1$. The topological distance between the neurons i and i', located at positions (y_1, y_2, y_3) and (y'_1, y'_2, y'_3) in the lattice space, is given by:

$$d(i, i') = \sqrt{(y_1 - y'_1)^2 + (y_2 - y'_2)^2 + (y_3 - y'_3)^2}. \tag{1}$$

Every neuron i has a prototype vector $\mathbf{w}_i$ which represents a cluster of input samples. Please note that $\mathbf{w}_i \in \mathbb{R}^D$, where D is the dimension of the input space. At time step n, a new sample $\mathbf{x}(n)$ is presented to the network, and a winner neuron is declared:

$$\text{Winner } (\mathbf{x}(n)) = \arg \min_{j \in \{1,\ldots,M\}} \|\mathbf{x}(n) - \mathbf{w}_j(n)\|, \tag{2}$$

where a tie breaking criterion must be defined. Then the prototypes of all the units are adjusted, for $i \in \{1, \ldots, M\}$:

$$\mathbf{w}_i(n+1) =$$
$$= \mathbf{w}_i(n) + \eta(n) \Lambda(i, \text{Winner}(\mathbf{x}(n)))(\mathbf{x}(n) - \mathbf{w}_i(n)), \tag{3}$$

where $\eta(n)$ is a decaying learning rate and the neighborhood function Λ varies with the time step n and depends on a decaying *neighborhood radius* $\Delta(n)$:

$$\eta(n+1) \leq \eta(n), \tag{4}$$

$$\Lambda(i, \text{Winner}(\mathbf{x}(n))) = \exp\left(-\left(\frac{d(i, \text{Winner}(\mathbf{x}(n)))}{\Delta(n)}\right)^2\right), \tag{5}$$

$$\Delta(n+1) \leq \Delta(n). \tag{6}$$

The receptive field of neuron i, i.e. the region of the input space which is represented by i, is defined as:

$$F_i = \left\{\mathbf{x} \in \mathbb{R}^D \mid i = \text{Winner}(\mathbf{x})\right\}. \tag{7}$$

Self-organizing maps are unsupervised learning neural networks which perform a vector quantization-type process. The performance of a map for this task is commonly measured by the mean squared error [4, 13, 19, 32]:

$$MSE = \frac{1}{K} \sum_{k=1}^{K} \min_{j \in \{1,\ldots,M\}} \|\mathbf{x}_k - \mathbf{w}_j\|^2, \tag{8}$$

where K is the number of input samples.

Next we see how SOMs can be linked to the minimization of the MSE by their energy function.

2.2 Energy function

The theoretical basis for the SOM algorithm given by Kohonen [9, 38] considers an energy measure,

$$\mathcal{E} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{M} I(\mathbf{x}_k \in F_i) \sum_{j=1}^{M} \Lambda(i,j) \|\mathbf{x}_k - \mathbf{w}_j\|^2, \tag{9}$$

where Λ is the neighborhood function, and I is the indicator function,

$$I\left(\text{Condition}\right) = \begin{cases} 0 & \text{iff Condition is false} \\ 1 & \text{iff Condition is true.} \end{cases} \tag{10}$$

Then the energy $\mathcal{E}$ is minimized by means of stochastic approximation (Robbins-Monro method). In order to guarantee the almost sure convergence of the algorithm, the step size $\gamma\left(t\right)$ of the Robbins-Monro method (which corresponds to the learning rate in the SOM) must verify the following conditions [28]:

$$\sum_{n=1}^{\infty} \gamma\left(n\right) = \infty, \tag{11}$$

$$\sum_{n=1}^{\infty} \left(\gamma\left(n\right)\right)^2 < \infty. \tag{12}$$

This is typically achieved by selecting.

$$\gamma\left(n\right) = \frac{a}{t+b}, \tag{13}$$

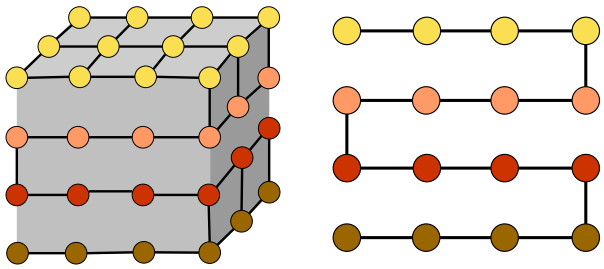
where a and b are suitable constants, as seen in [20], for example.

3. The role of the lattice dimensionality

This section studies the effect of the lattice dimensionality on the SOM from several points of view. First of all, the energy function of the SOM is decomposed to show the differences among topologies from a network level perspective (Subsection 3.1). Then a method to assess the optimal neuron configurations produced by a SOM energy function is developed; it focuses on the adaptation of the neurons to the local features of the input distribution (Subsection 3.2). Thirdly the implications of the intrinsic structure of the input distribution are explored (Subsection 3.3). Finally, the behavior of the network near a local minimum of the energy function is considered (Subsection 3.4).

3.1 Energy function decomposition

Here the energy function of a SOM is to be decomposed according to its lattice dimensionality. First of all, a way to obtain a SOM of a lower dimensionality is considered (Fig. 1). Given a 3D lattice of size $a \times b \times c$, we can obtain a 2D lattice with the same number of neurons by joining side by side the c rectangular lattices (sheets) of size $a \times b$ which form the original 3D lattice, as shown in Fig. 1a. This yields a 2D lattice of size $a \times bc$. Moreover, we can obtain a 1D lattice with the same number of neurons by connecting one after the other the a rows of size bc which form the 2D lattice, as shown in Fig. 1b. This yields a 1D lattice of size $1 \times abc$.



(a) (b)

Fig. 1 *Lattice dimension reduction: (a) obtaining a 2D lattice of size* 3×16 *from a 3D lattice of size* $3 \times 4 \times 4$*, (b) obtaining a 1D lattice of size* 1×16 *from a 2D lattice of size* 4×4*. The topological connections which are kept are shown as black lines.*

Let us note the topological distances for the three lattices d_3, d_2, and d_1, respectively. The corresponding neighborhood functions will be noted Λ_3, Λ_2, and Λ_1, where we assume that the neighborhood radius $\Delta(n)$ is the same for the three lattices at all time steps n. Now, from the definition of the lattices the following relations hold for all pairs of neurons i, j:

$$d_3(i,j) \leq d_2(i,j), \tag{14}$$

$$d_2(i,j) \leq d_1(i,j). \tag{15}$$

For reasons that will become clear later, the topological distances for a null topology and a fully connected topology are also defined:

$$d_0(i,j) = \begin{cases} 0 & \text{iff } i = j \\ \infty & \text{iff } i \neq j, \end{cases} \tag{16}$$

$$d_{Full}(i,j) = 0. \tag{17}$$

From (16) and (17) we get:

$$d_{Full}(i,j) \leq d_3(i,j), \tag{18}$$

$$d_1(i,j) \leq d_0(i,j). \tag{19}$$

Equations (14), (15), (18) and (19) imply that:

$$\Lambda_{Full}(i,j) \geq \Lambda_3(i,j), \tag{20}$$

$$\forall h \in \{1,2,3\},\ \Lambda_h(i,j) \geq \Lambda_{h-1}(i,j). \tag{21}$$

Next we can decompose the energy function of the 3D map as follows:

$$\mathcal{E}_{3D} = E_0 + E_1 + E_2 + E_3, \tag{22}$$

$$E_0 = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{M} I\left(\mathbf{x}_k \in F_i\right)\sum_{j=1}^{M}\Lambda_0\left(i,j\right)\left\|\mathbf{x}_k - \mathbf{w}_j\right\|^2, \tag{23}$$

$$\forall h \in \{1,2,3\},\ E_h =$$

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{M} I\left(\mathbf{x}_k \in F_i\right)\sum_{j=1}^{M}\left(\Lambda_h\left(i,j\right) - \Lambda_{h-1}\left(i,j\right)\right)\left\|\mathbf{x}_k - \mathbf{w}_j\right\|^2. \tag{24}$$

From (16) and (21) we obtain:

$$\forall h \in \{0,1,2,3\},\ E_h \geq 0. \tag{25}$$

Then, from (8) and (23):

$$E_0 = MSE. \tag{26}$$

Finally, the energy functions for the 1D SOM and the 2D SOM can be decomposed as follows:

$$\mathcal{E}_{1D} = E_0 + E_1, \tag{27}$$

$$\mathcal{E}_{2D} = E_0 + E_1 + E_2. \tag{28}$$

From the preceding it can be inferred that a 1D SOM has an energy function which is closer to the MSE than a 2D SOM, and that the energy function of a 3D SOM is even farther from the MSE than that of a 2D SOM. It is also interesting to remember that the competitive learning neural network, which corresponds to the null topological distance d_0, minimizes the MSE. So we can sort the models according to the importance of the MSE in their energy functions: competitive learning (maximum importance), 1D SOM, 2D SOM and 3D SOM (minimum importance).

Some insight about the effect of adding topological connection terms to the energy function can be obtained if we consider a fully connected topology, which corresponds to this energy:

$$\mathcal{E}_{Full} = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{M} I\left(\mathbf{x}_k \in F_i\right)\sum_{j=1}^{M}\Lambda_{Full}\left(i,j\right)\left\|\mathbf{x}_k - \mathbf{w}_j\right\|^2,$$

$$= E_0 + E_1 + E_2 + E_3 + E_{Full}.$$

$$= \frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{M}\left\|\mathbf{x}_k - \mathbf{w}_j\right\|^2, \tag{29}$$

where

$$E_{Full} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{M} I\left(\mathbf{x}_k \in F_i\right) \sum_{j=1}^{M} \left(\Lambda_{Full}\left(i,j\right) - \Lambda_3\left(i,j\right)\right) \left\|\mathbf{x}_k - \mathbf{w}_j\right\|^2. \tag{30}$$

It must be noted that (20) and (30) imply that:

$$E_{Full} \geq 0. \tag{31}$$

Next we prove that that $\mathcal{E}_{Full}$ is minimized for $\mathbf{w}_j = E\left[\mathbf{x}\right]$, where $E\left[\mathbf{x}\right]$ stands for the mean vector of the overall input distribution:

Theorem 1. The global minimum of $\mathcal{E}_{Full}$ is $\mathbf{w}_j = E\left[\mathbf{x}\right]$.

Proof. To see that $\mathcal{E}_{Full}$ is minimized for $\mathbf{w}_j = E\left[\mathbf{x}\right]$ we set $w_j = E\left[\mathbf{x}\right] + \delta_j$ and evaluate $\mathcal{E}_{Full}$. We obtain

$$\mathcal{E}_{Full} = \mathcal{E}_{Full}|_{w_j = E[x]} + Z + \Delta, \tag{32}$$

where

$$Z = \frac{2}{K} \sum_{k=1}^{K} \sum_{j=1}^{M} (\mathbf{x}_k - E[x]) \cdot \delta_j = 0 \tag{33}$$

and

$$\Delta = \sum_{j=1}^{M} \left\|\delta_j\right\|^2 \geq 0 \tag{34}$$

so the theorem follows.

This theorem means that for a fully connected topology the energy is minimized if and only if all the prototypes collapse to the global mean of the input distribution. From these considerations, we conclude that the more topological connections that are added to the map, the less importance that it gives to the minimization of the MSE and the stronger the attraction of the learned prototypes towards the global mean of the input distribution.

However, this does not imply that the plain minimization of the MSE, which corresponds to a purely competitive neural network, would yield the best results in terms of MSE. As known, the absence of cooperation among neurons leads to local minima corresponding to dead neurons, i.e. units with little or no samples in their receptive fields. It is worth noting that these local minima are removed by the introduction of the E_1 term, since this connects all neurons together so that a dead neuron is no longer a local minimum of the energy function. Consequently, the extra terms E_2, E_3,... offer no further advantage in this sense.

Here the overall goal of the network (its energy function) has been studied from a global point of view. That is, the adaptation to the local features of the input distribution has not been considered. In the next subsection, the energy function is assessed from a local viewpoint.

3.2 Energy function assessment

The energy function decomposition that has been carried out in Subsection 3.1 does not give specific information about the distribution of the local minima of the energy functions associated to different topologies. This task can not be accomplished by comparing the energy functions directly, because an energy function has MD parameters (the components of the M prototype vectors), so it is defined on $\mathbb{R}^{MD}$ which is a very high dimensional space. Even worse, since the neighborhood radius varies during the learning process the energy function also varies, which implies that the study of any particular case of the energy function does not apply to the overall learning process. Of course, one can always compare the resulting MSE for different topologies, but the MSE is a global scalar performance measure, so it conveys no information about the way that the networks adapt to the input distribution.

Let us consider a hypercube $V(\mathbf{x})$ in the input space centered around $\mathbf{x} \in \mathbb{R}^D$ where $V(\mathbf{x}) \subset \mathbb{R}^D$, and let $Vol(V(\mathbf{x}))$ be its D-dimensional volume. Let $n(V(\mathbf{x}))$ be the expectation of the number of neurons that are inside $V(\mathbf{x})$ when the SOM algorithm finishes, given an input distribution and a certain choice of the SOM learning parameters. Then we define the neuron density at the point $\mathbf{x} \in \mathbb{R}^D$ as follows:

$$\rho(\mathbf{x}) = \lim_{Vol(V(\mathbf{x}))\to 0} \frac{n(V(\mathbf{x}))}{M \cdot Vol(V(\mathbf{x}))}. \tag{35}$$

Now, if we note $f_{\mathbf{w}_i}(\mathbf{x})$ the probability density of the final value of the i-th prototype vector of the SOM, we have:

$$\rho(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} f_{\mathbf{w}_i}(\mathbf{x}). \tag{36}$$

It is also worthwhile noting that the integral of the neuron density over the entire input space is one:

$$\int_{\mathbb{R}^D} \rho(\mathbf{x})\, d\mathbf{x} = 1. \tag{37}$$

We propose the use of the neuron density ρ as a tool to examine the optimal neuron configurations corresponding to the local features of the input data. Neuron density has been previously studied in literature. The well known *SOM magnification factor* result states that $\rho(\mathbf{x})$ weakly converges when $M \to \infty$ to the following probability distribution [22, 28]:

$$\hat{\rho}(\mathbf{x}) = \frac{(p(\mathbf{x}))^{\frac{D}{D+2}}}{\int_{\mathbb{R}^D} (p(\mathbf{x}))^{\frac{D}{D+2}}}, \tag{38}$$

where $p(\mathbf{x})$ is the input distribution.

However, the result (38) is not directly applicable to practical SOMs because the number of neurons M is a finite number. On the other hand, the detailed examination of $\rho(\mathbf{x})$ for a given SOM with a certain set of learning parameters and network topology can give some insights about the adaptation of the network to

the input distribution at hand. High values of $\rho(\mathbf{x})$ mean that prototype vectors usually finish near $\mathbf{x}$, which implies that putting a prototype near $\mathbf{x}$ usually leads to a lower value of the energy function of the SOM. Consequently, the local maxima of $\rho(\mathbf{x})$ correspond to neuron locations associated to minimal energy configurations of a SOM. This allows to indirectly evaluate the difference between the MSE and the energy function associated to a SOM by comparing the neuron density of competitive learning neural network and that of the SOM. The closer the neuron density of the competitive network to that of the SOM, the more similar their energies. A fundamental advantage of this methodology is that the energy function is defined over $\mathbb{R}^{MD}$, while the neuron density is defined over $\mathbb{R}^{D}$. Hence the neuron density is easier to assess than the energy function, and it is even possible to plot it for $D = 2$. Moreover, the neuron density carries information about the whole learning process, while the energy function varies over this process, as mentioned before.

3.3 Intrinsic dimensionality of the input

In this subsection a study of the effect of the intrinsic dimensionality of the input distribution is carried out. A simple distribution is chosen for this purpose, so that a full mathematical treatment is possible. In particular, the uniform distribution on a D-dimensional box (also called orthotope) is considered, with the following input probability density:

$$p(\mathbf{x}) = \prod_{h=1}^{D} U(x_h, \alpha_h, \beta_h), \tag{39}$$

where α_h, β_h are the lower and upper limits of the box in the h-th dimension, and U stands for the univariate uniform distribution. It must be pointed out that it is assumed that the box is aligned with the coordinate axes, without loss of generality. Let λ_h be the length of the box in the h-th dimension,

$$\lambda_h = \beta_h - \alpha_h. \tag{40}$$

It is also assumed without loss of generality that

$$\forall h \in \{1, \ldots, D-1\},\ \lambda_h \geq \lambda_{h+1} \tag{41}$$

that is, the first principal direction (the direction with the highest variance) is dimension 1, and so on. Next the performances of 1D, 2D and 3D SOMs over this distribution are compared.

3.3.1 1D versus 2D SOMs

For a 1D SOM the globally optimum configuration of the map is one where the neurons are evenly distributed over the first principal direction, provided that the difference between λ_1 and λ_2 is large enough that no curved configurations of the map are better, and neglecting the border effects near the limits of the box. This means that each receptive field is a box with length $\frac{\lambda_1}{M}$ over the first dimension, and λ_h over the remaining dimensions, $h \in \{2, \ldots, D\}$. The mean squared error

corresponding to this optimal configuration can be computed as the sum of the variances in a receptive field over the D dimensions of the input space. This amounts to the sum of the variance of a uniform distribution for each of the D dimensions:

$$MSE_{1D} = \frac{1}{12}\left(\frac{\lambda_1}{M}\right)^2 + \frac{1}{12}\sum_{h=2}^{D}\lambda_h^2. \tag{42}$$

On the other hand, for a 2D SOM with the same number of neurons and a square topology, i.e. a $\sqrt{M}\times\sqrt{M}$ topology, the globally optimum configuration of the map is one where the neurons are evenly distributed over the first and second principal directions, provided that the difference between λ_2 and λ_3 is large enough that no curved configurations of the map are better, and neglecting again the border effects near the limits of the box. This implies that each receptive field is a box with length $\frac{\lambda_1}{\sqrt{M}}$ over the first dimension, $\frac{\lambda_2}{\sqrt{M}}$ over the second dimension, and λ_h over the remaining dimensions, $h \in \{3,\ldots,D\}$. The associated mean squared error is:

$$MSE_{square} = \frac{1}{12}\left(\frac{\lambda_1}{\sqrt{M}}\right)^2 + \frac{1}{12}\left(\frac{\lambda_2}{\sqrt{M}}\right)^2 + \frac{1}{12}\sum_{h=3}^{D}\lambda_h^2. \tag{43}$$

Under these conditions, the MSE performance of the 1D SOM is better than that of the 2D SOM with square topology iff the following condition holds:

$$MSE_{1D} < MSE_{square}, \tag{44}$$

$$\frac{1}{12}\frac{\lambda_1^2}{M^2} + \frac{1}{12}\lambda_2^2 < \frac{1}{12}\frac{\lambda_1^2}{M} + \frac{1}{12}\frac{\lambda_2^2}{M}, \tag{45}$$

which is satisfied for $\lambda_1 \gg \lambda_2$, i.e. when the intrinsic dimensionality of the input distribution is one.

If the analysis is generalized to arbitrary 2D rectangular topologies of size $a\times\frac{M}{a}$, the receptive fields are boxes with sizes $\frac{\lambda_1}{a}$ and $\frac{a\lambda_2}{M}$ over the first and second dimensions

$$MSE_{rect} = \frac{1}{12}\left(\frac{\lambda_1}{a}\right)^2 + \frac{1}{12}\left(\frac{a\lambda_2}{M}\right)^2 + \frac{1}{12}\sum_{h=3}^{D}\lambda_h^2. \tag{46}$$

The rectangular topology which minimizes MSE_{rect} is obtained as follows:

$$\frac{\partial MSE_{rect}}{\partial a} = \frac{1}{12}\left(-2\frac{\lambda_1^2}{a^3} + \frac{2a\lambda_2^2}{M^2}\right) = 0, \tag{47}$$

$$2\frac{\lambda_1^2}{a^3} = \frac{2a\lambda_2^2}{M^2}, \tag{48}$$

$$a = \sqrt{\frac{M\lambda_1}{\lambda_2}}. \tag{49}$$

If the intrinsic dimensionality is one, i.e. $\lambda_1 \gg \lambda_2$, then the solution for a grows without limit. Consequently the optimal map size is $M \times 1$, so we recover the 1D SOM. On the other hand, if $\lambda_1 = \lambda_2$ (the two first principal directions have the same variances) the optimum is attained for $a = \sqrt{M}$, which is the square topology. It can be concluded from (49) that the optimal topology becomes more elongated as the ratio between the variance of the first principal direction and the second principal direction grows.

3.3.2 2D versus 3D SOMs

For a 3D SOM with M neurons and a cubic topology, i.e. a $\sqrt[3]{M} \times \sqrt[3]{M} \times \sqrt[3]{M}$ topology, the globally optimum configuration of the map is one where the neurons are evenly distributed over the first, second and third principal directions, provided that the difference between λ_3 and λ_4 is large enough that no curved configurations of the map are better, and neglecting the border effects near the limits of the box. This implies that each receptive field is a box with lengths $\frac{\lambda_1}{\sqrt[3]{M}}$, $\frac{\lambda_2}{\sqrt[3]{M}}$ and $\frac{\lambda_3}{\sqrt[3]{M}}$ over the first three dimensions, and λ_h over the remaining dimensions, $h \in \{4, \ldots, D\}$. The associated mean squared error is:

$$MSE_{cubic} =$$

$$\frac{1}{12\sqrt[3]{M^2}} \left(\lambda_1^2 + \lambda_2^2 + \lambda_3^2\right) + \frac{1}{12} \sum_{h=4}^{D} \lambda_h^2. \tag{50}$$

Under these conditions, the MSE performance of the 2D SOM with square topology is better than that of the 3D SOM with cubic topology iff the following condition holds:

$$MSE_{square} < MSE_{cubic}, \tag{51}$$

$$\frac{1}{12} \left(\frac{\lambda_1^2}{M} + \frac{\lambda_2^2}{M} + \lambda_3^2\right) < \frac{1}{12M^{2/3}} \left(\lambda_1^2 + \lambda_2^2 + \lambda_3^2\right). \tag{52}$$

which is satisfied for $\lambda_2 \gg \lambda_3$, i.e. when the intrinsic dimensionality of the input distribution is lower than three. This result is in agreement with the one that was obtained for 1D and 2D SOMs, so that it can be inferred that a lattice dimensionality that matches the intrinsic dimensionality of the input yields the best results in terms of MSE.

For a rectangular cuboid of size $a \times b \times \frac{M}{ab}$ we are led to

$$MSE_{rect} =$$

$$\frac{1}{12} \left(\frac{\lambda_1}{a}\right)^2 + \frac{1}{12} \left(\frac{\lambda_2}{b}\right)^2 + \frac{1}{12} \left(\frac{ab\lambda_3}{M}\right)^2 + \frac{1}{12} \sum_{h=4}^{D} \lambda_h^2. \tag{53}$$

To find the critical points we proceed as before:

$$\frac{\partial MSE_{rect}}{\partial a} = \frac{1}{12}\left(-2\frac{\lambda_1^2}{a^3} + \frac{2ab^2\lambda_3^2}{M^2}\right) = 0, \tag{54}$$

$$\frac{\partial MSE_{rect}}{\partial b} = \frac{1}{12}\left(-2\frac{\lambda_2^2}{b^3} + \frac{2ba^2\lambda_3^2}{M^2}\right) = 0. \tag{55}$$

Solving for a or b in one equation and substituting in the other one we find:

$$a = \sqrt[3]{\frac{M\lambda_1^2}{\lambda_2\lambda_3}} \text{ and } b = \sqrt[3]{\frac{M\lambda_2^2}{\lambda_1\lambda_3}}. \tag{56}$$

When the intrinsic dimensionality of the input is 1 or 2, i.e., for $\lambda_2 \gg \lambda_3$, we also have $\lambda_1 \gg \lambda_3$ and both a and b become as large as we wish. When the three first principal directions have equals variances, $\lambda_1 = \lambda_2 = \lambda_3$, the cubic solution $a = b = \sqrt[3]{M}$ is the optimum one.

3.4 Local analysis

The study of energy minimization carried out in previous sections showed dimension-depending features of SOM. A general analysis of its behavior on an iteration-by-iteration basis seems impossible. Nevertheless, in this section and under some hypotheses, we shall accomplish such an investigation, providing an alternative confirmation of the characteristics of SOM previously described.

Assume the network has reached a local minimum for MSE. Then a number of samples equal to the number of neurons, $K = M$, are presented in the order $\mathbf{x}_1, \ldots, \mathbf{x}_K$. We further impose the condition that $\mathbf{x}_j$ belongs to the receptive field of neuron j, F_j.

To simplify, we consider a constant learning rate η and a constant neighborhood radius $\Delta = 1$ throughout the iterations. Moreover, because the learning rule (3) is continuous on η, we may also assume that the winner neuron on iteration k is the neuron $j = k$ by taking η small enough. It is straightforward then that after K iterations we have

$$\mathbf{w}_i(K) = \alpha_{i,0}\mathbf{w}_i(0) + \sum_{k=1}^{K} \beta_{i,k}\alpha_{i,k}\mathbf{x}_k, \tag{57}$$

where $\mathbf{w}_i(0) = \mathbf{w}_i = \mathbf{x}_i$, $\beta_{i,k} = \eta\Lambda(i,k)$ and $\alpha_{i,k} = \prod_{j=k+1}^{K}(1-\beta_{i,k})$. Note that $\beta_{i,i} = \eta$ and the strong order-depending behavior.

If we let the distance in the lattice grow without limit, $d \to \infty$, then we have

$$\beta_{i,k} \to \begin{cases} \eta & k = i \\ 0 & k \neq i \end{cases} \text{ and } \alpha_{i,k} \to \begin{cases} 1 & k \geq i \\ 1-\eta & k < i \end{cases}$$

and Eq. (57) becomes

$$\mathbf{w}_i(K) = (1-\eta)\mathbf{w}_i(0) + \eta\mathbf{x}_i.$$

This shows that in this setup vector quantization does remain close to the local minimum for MSE. On the other hand, if we let d decrease without limit, $d \to 0$, then we have $\beta_{i,k} \to \eta$, $\alpha_{i,k} \to 1-\eta$ and Eq. (57) becomes

$$\mathbf{w}_i(K) = (1-\eta)^K \mathbf{w}_i(0) + \sum_{k=1}^{K} \eta(1-\eta)^{K-k}\mathbf{x}_i.$$

Note that, because $0 < 1-\eta < 1$, the expression $(1-\eta)^l$ approaches 0 as l grows. So for small lattice distances, the network forgets the early samples and the original state $\mathbf{w}_i(0)$ and moves towards the late samples.

As we observed in Section 3.1, the distance of the lattice d is inversely proportional to the dimension of the lattice D. So the conclusions outlined in the paragraph above for large or small d also apply to small or large D. Anyhow, to examine in a more explicit way the impact of lattice dimensionality on Eq. (57) we study the coefficients in this expression. We start by computing the average value β of $\beta_{i,k}$ for k different from i. Notice that $\beta_{i,k}$ depends only on the distance $d(i,k)$ from i to k. So, if we write M_l for the number of neurons at distance l from the neuron i (hence $M_0 = 1$) and we let r be the maximum distance to i we have

$$\beta = \frac{1}{M_1 + M_2 + \ldots + M_r} \sum_{l=1}^{r} M_l \eta \Lambda(l),$$

where $\Lambda(l) = \exp(-l^2)$. The dimension of the lattice provides the estimates $M_l \approx M_1 l^{D-1}$ and $M \approx r^D$ which gives finally

$$\beta \approx \eta \frac{\sum_{l=1}^{r} l^{D-1} \exp(-l^2)}{\sum_{l=1}^{r} l^{D-1}}. \tag{58}$$

This number β is a increasing function of D. For instance, let us consider the following values of M_l for $D = 1, 2, 3$ and small l:

D	M_0	M_1	M_2	M_3	M_4
1	1	2	2	2	2
2	1	8	16	24	32
3	1	26	104	234	416

Then for fixed number of neurons $M = 27$ we have:

D	r	β/η
1	13	0.0297
2	≈ 2	0.1348
3	1	0.3679

The coefficient of $\mathbf{w}_i(0)$ in (57) can be approximated by $(1-\beta)^K$. Hence, this coefficient is decreasing in D. The ratio of the coefficients of $\mathbf{x}_k$ $(k \neq i)$ and of $\mathbf{x}_i$ is close to

$$\frac{\beta(1-\beta)^{K-k}}{\eta(1-\beta)^{K-i}} = \frac{\beta}{\eta}(1-\beta)^{i-k}. \tag{59}$$

This function is increasing in β for small enough values of β, and for this it suffices to choose η small enough too. Hence, this ratio is an increasing function of D.

To sum up, the larger the dimension D the less spread the distances among the neurons of the lattice. This causes the neuron i tends to forget its initial state $\mathbf{w}_i(0)$, suffers larger influence of samples which do not belong to its receptive field F_i and move away from the local minimum. In contrast, for low dimensions D, the neurons of the lattice are more evenly distributed proportionally to its distance to a fixed neuron. In this case, the network tends to stay near the local minimum for MSE.

4. Experiments

In order to assess the proposed grid topologies, two kinds of experiments have been designed[1]. Synthetic 2D and 3D datasets have been used to study the neuron density under different conditions (Subsection 4.2). Then the unsupervised clustering performance of the methods has been tested over image and machine learning data (Subsection 4.3). Before they are presented, the elements of the experimental design which are common to all the experiments are specified (Subsection 4.1).

4.1 Experimental design

The proposals have been implemented in Matlab with the most time consuming sections coded in C language, and they have been run on a single core of a 3GHz CPU with 64 bit architecture and 8GB of RAM.

Each prototype vector has been initialized to a training sample chosen uniformly at random from the training set. Next we detail the parameter selection strategy for the experiments. Let N be the overall number of time steps of the training process. It has been set to $N = 100,000$ for all the experiments. We have divided the training process into an ordering phase and a convergence phase with the same number of time steps, i.e. $\frac{N}{2}$ time steps each. During the ordering phase, the learning rate and the neighborhood radius experience a linear decay:

$$\eta(n) = \eta_0 \left(1 - \frac{n}{N}\right), \tag{60}$$

$$\Delta(n) = \Delta_0 \left(1 - \frac{n-1}{N}\right). \tag{61}$$

During the convergence phase, constant values have been used to carry out the fine tuning of the maps: $\eta(n) = \eta_c$, $\Delta(n) = \Delta_c$. Hence, the set of parameters to be chosen is: η_0, Δ_0, η_c, Δ_c. The parameter choice strategy has been different for the synthetic and real datasets, as specified next.

4.1.1 Synthetic datasets

For the 2D synthetic datasets (Subsection 4.2) competitive learning, 1D and 2D topologies have been compared. No 3D topologies have been tested, since it does

[1]The source code and demos of our proposal will be published in case of acceptance.

not make sense for 2D datasets. On the other hand, for the 3D synthetic datasets we have also carried out tests with 3D topologies. We have fixed $\eta_0 = 0.4$ and $\eta_c = 0.01$. Then for the 2D and 3D topologies we have chosen $\Delta_0 = \frac{\sqrt{M}}{32}$ and $\Delta_c = \frac{\sqrt{M}}{128}$. For the 1D topologies we have chosen $\Delta_0 = \frac{M}{4}$ and $\Delta_c = \frac{M}{64}$. The values have been selected by hand so as to obtain good adaptations of the networks to the synthetic input distributions; the adaptation has been assessed visually. The map sizes for 2D datasets have been $4 \times 4 \times 1$, $6 \times 6 \times 1$ and $8 \times 8 \times 1$ for 2D SOMs, and the same numbers of neurons ($M = 16,\ 36,\ 64$) for competitive learning and 1D SOMs. For the 3D datasets we have tested 2D map sizes of $8 \times 8 \times 1$ and $10 \times 15 \times 1$, 3D maps sizes of $4 \times 4 \times 4$ and $5 \times 5 \times 6$, and the same numbers of neurons for competitive learning and 1D SOMs ($M = 64,\ 150$).

4.1.2 Real datasets

For the real datasets, which have $D \geq 3$ it is not advisable to select the parameters by hand, since it is difficult to plot the resulting configurations of the maps. Consequently, we have considered the Mean Squared Error (8) as the objective function to be minimized. The Nelder-Mead optimization method John A. and R. [25] has been used to carry out the parameter optimization, which is quite robust with respect to noise in the objective function. To this end we split the available set of samples into a training set (90% of the data) and a validation set (the remaining 10%). The map is trained with the training set and then the MSE is evaluated over the validation set. In this set of experiments competitive learning, 1D, 2D and 3D topologies have been considered. For each dataset, simulations have been repeated for map sizes of $64 \times 1 \times 1$, $150 \times 1 \times 1$, $216 \times 1 \times 1$, $294 \times 1 \times 1$, and $729 \times 1 \times 1$ neurons (1D); $8 \times 8 \times 1$, $10 \times 15 \times 1$, $12 \times 18 \times 1$, $14 \times 21 \times 1$ and $27 \times 27 \times 1$ neurons (2D); and $4 \times 4 \times 4$, $5 \times 5 \times 6$, $6 \times 6 \times 6$, $7 \times 7 \times 6$ and $9 \times 9 \times 9$ neurons (3D). Please note that the same numbers of neurons have been used for 1D, 2D and 3D maps: 64, 150, 216, 294 and 729, respectively. The numbers of neurons for the competitive learning networks have been the same.

4.2 Neuron density

In this set of experiments the goal is to compare the neuron densities produced by competitive learning, 1D SOMs and 2D SOMs. Since competitive learning corresponds to plain MSE minimization, neuron densities similar to that of competitive learning are expected to be associated with low values of the MSE.

Three 2D and two 3D input datasets have been considered (see Fig. 2). For each 2D dataset, number of neurons and model, 1,000 runs have been carried out, while 10,000 runs have been executed for 3D datasets because the higher input dimension requires a higher number of samples to estimate ρ accurately. Then the final prototype vectors have been fed to the FIGTree implementation [43, 44] of the Improved Fast Gauss Transform (IFGT) with bandwidth $\sigma = 0.02$ in order to produce an accurate estimation of the neuron density ρ.

The results for 2D datasets are shown in Figs. 3 to 5. It can be noticed that the neuron density for the competitive learning smooths out as the number of neurons grows. It tends to follow the input density, although there are spots with higher

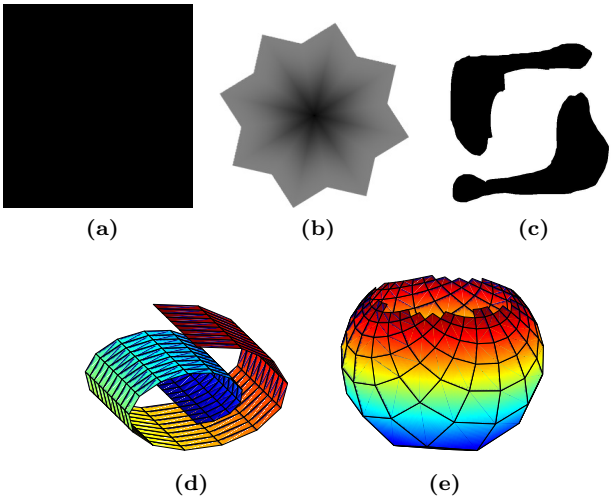


Fig. 2 *Synthetic datasets: (a) square, (b) star, (c) two shapes, (d) Swiss roll, (e) punctured sphere. For the star dataset, darker tones mean higher input density.*

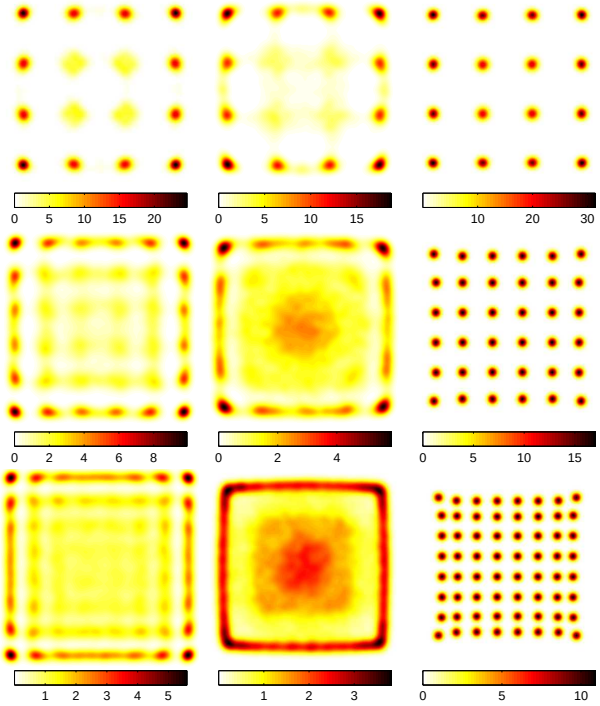


Fig. 3 *Neuron densities for the square dataset. From left to right: competitive learning, 1D SOM and 2D SOM. From top to bottom:* $M = 16$, $M = 36$ *and* $M = 64$. *The neuron density keys are shown below each subfigure.*

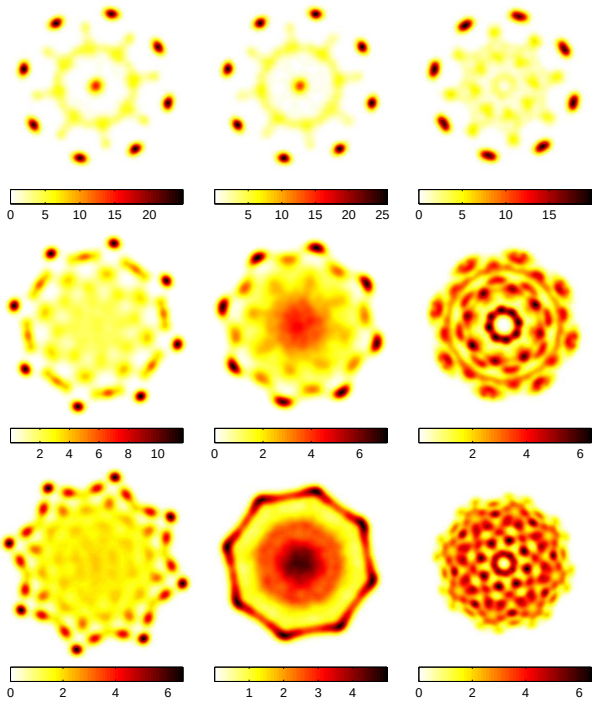


Fig. 4 *Neuron densities for the star dataset. From left to right: competitive learning, 1D SOM and 2D SOM. From top to bottom:* $M = 16$, $M = 36$ *and* $M = 64$. *The neuron density keys are shown below each subfigure.*

ρ at the corners of the input distribution. This means that MSE is significantly lowered whenever a neuron is placed in one of these spots. The 1D SOM follows the same trend, and it smooths out even faster than competitive learning. On the other hand, the 2D SOM shows a completely different behavior. Either ρ concentrates on discrete spots, even for high values of M (Figs. 3 and 5), or it develops complex patterns which have nothing to do with the input distribution and are due to the 2D lattice constraint (Fig. 4).

The obtained neuron densities for 3D datasets are depicted in Figs. 6 and 7. It is worth noting that the Swiss roll dataset has a uniform density over all the manifold, while the punctured dataset has a higher density in the points of the manifold which are closer to the hole. Again competitive learning follows the input density, with high ρ spots in the extreme points of the input distribution. 1D SOMs also follow the input, although they tend to place some neurons in the inner region of the roll which has no input samples, and in the hole of the punctured sphere. 3D SOMs offer even better adaptation to the input, even if they still place a few neurons in zero density regions which lie among high input density regions. Finally, 2D SOMs yield the worst performance, since they spread the neurons all over the zero input density regions of the interior of the Swiss roll, and they fail to place neurons in the lower input density regions of the punctured sphere.

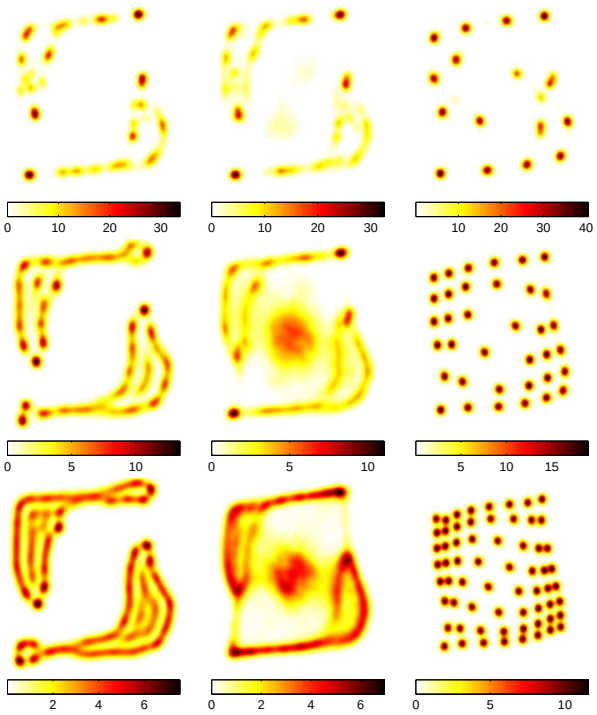


Fig. 5 *Neuron densities for the two shapes dataset. From left to right: competitive learning, 1D SOM and 2D SOM. From top to bottom:* $M = 16$, $M = 36$ *and* $M = 64$. *The neuron density keys are shown below each subfigure.*

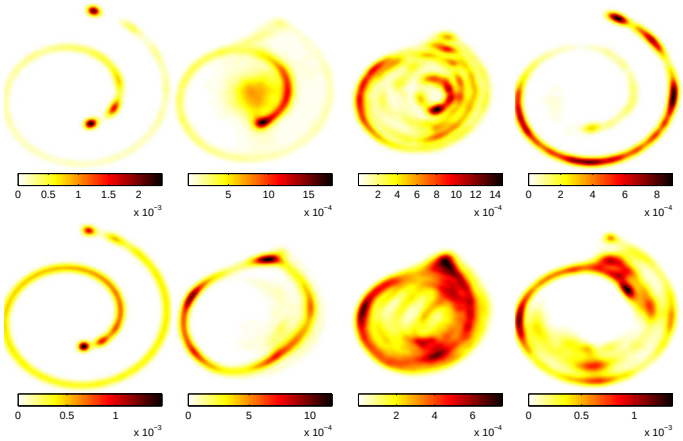


Fig. 6 *Neuron densities for the SwissRoll dataset over the plane which splits the roll into two halves. From left to right: competitive learning, 1D SOM, 2D SOM and 3D SOM. From top to bottom:* $M = 64$ *and* $M = 150$. *The neuron density keys are shown below each subfigure.*

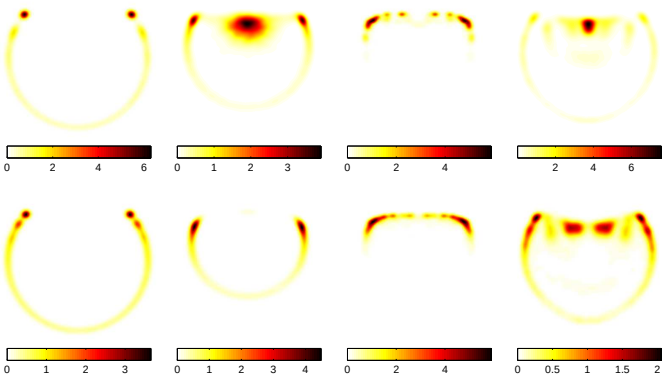


Fig. 7 *Neuron densities for the punctured sphere dataset over a plane which splits the dataset into two halves. From left to right: competitive learning, 1D SOM, 2D SOM and 3D SOM. From top to bottom:* $M = 64$ *and* $M = 150$*. The neuron density keys are shown below each subfigure.*

It can be concluded that 1D SOMs distribute their neurons in a way that follows the input distribution more closely than 2D SOMs, which are strongly constrained by the 2D lattice. On the other hand, 3D SOMs offer a good performance in the 3D datasets, which can be related to the match of the input space dimension and the map lattice dimension.

4.3 Unsupervised clustering

A natural application of self organizing maps is unsupervised clustering, since these maps do not need previously labeled training samples in order to obtain a meaningful clustering of an input distribution [3, 18, 27, 34, 37, 46]. Hence, several datasets have been selected from two domains: color images and machine learning.

We have chosen six well known benchmark images from the USC-SIPI Image Database [5], which are shown in Fig. 8. All of them have size 512×512 pixels except House, which is 256×256 pixels. The pixel values for each RGB channel have a precision of 8 bits, and they lie in the range $[0, 255]$. Our second application domain deals with machine learning datasets from the UCI Repository of Machine Learning Databases [1]. The considered datasets are listed in Tabs. I (small size datasets) and II (large size datasets).

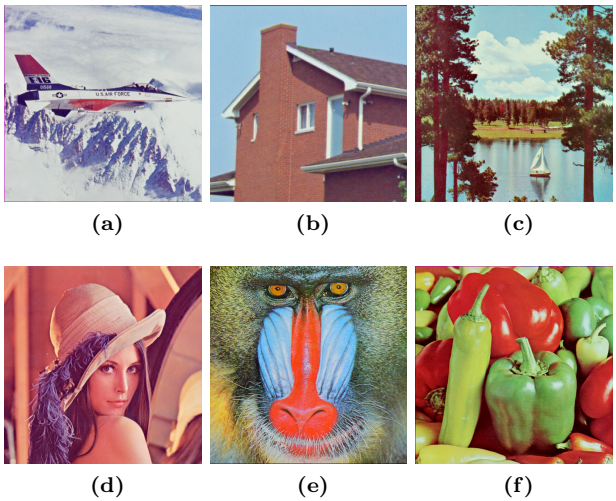
Three quantitative performance measures have been obtained for this set of experiments: Mean Squared Error (MSE, Eq. 8), Mean Silhouette Value (MSV) and Mean Tied Rank (MTR), that we discuss below.

The silhouette value is specifically designed to assess the quality of an unsupervised clustering [6, 33, 36]. Let σ_k be the average distance from sample $\mathbf{x}(k)$ to the other points in its own cluster, and $\sigma_{j,k}$ the average distance from $\mathbf{x}(k)$ to points in another cluster j. The silhouette value for a sample $SV(\mathbf{x}(k)) \in [-1, 1]$ and the Mean Silhouette Value (higher is better) are given by:

$$SV\left(\mathbf{x}\left(k\right)\right) = \frac{-\sigma_k + \min_j \sigma_{j,k}}{\max\left\{\sigma_k, \min_j \sigma_{j,k}\right\}}, \tag{62}$$

$$MSV = \frac{1}{K}\sum_{k=1}^{K} SV\left(\mathbf{x}\left(k\right)\right). \tag{63}$$

In order to compare the topology preservation ability of a self-organizing map, the topographic error is commonly used [4, 8, 14, 30]. However, it can not be used to compare maps with different topologies, because the TE depends critically on



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Fig. 8 *Benchmark color images: (a) F16, (b) House, (c) Lake, (d) Lena, (e) Mandrill, (f) Peppers.*

Dataset	D	# samples
BalanceScale	4	625
BreastCancerWisconsin	9	683
Cloud	10	2,048
Contraceptive	9	1,473
Dermatology	34	358
Glass	9	214
Haberman	3	306
HayesRoth	4	132
Liver	6	345
Pima	8	769
Vowel	10	990
Wine	13	178
Yeast	8	1,484

Tab. I *Small UCI benchmark datasets considered for unsupervised clustering.*

Dataset	D	# samples
CorelColorHistogram	32	68,039
CorelColorMoments	9	68,039
CorelCoocTexture	16	68,039
CorelLayoutHistogram	32	66,615
CoverType	10	581,010
Letter	16	20,000
MiniBooNE	50	130,063
SkinSegmentation	3	245,056

Tab. II *Large UCI benchmark datasets considered for unsupervised clustering.*

the number of immediate neighbors of each unit, and the number of immediate neighbors varies (4 neighbors for 2D topologies and 6 neighbors for 3D topologies).

In order to obtain a topology preservation measure which treats all possible map topologies on equal terms, the Mean Tied Rank (MTR) can be used [15]. For each test sample we compute the list of all the units of the map which are not the first best matching unit, sorted by topological distance to the first best matching unit. Then we compute the tied rank of the second best matching unit in this list. Finally, the MTR is defined as the mean of these tied ranks for all test data (lower is better):

$$MTR = \frac{1}{K} \sum_{k=1}^{K} \tau \left(\mathbf{x} \left(k \right) \right), \tag{64}$$

where $\tau \left(\mathbf{x} \left(k \right) \right)$ stands for the tied rank of the second best matching unit for sample $\mathbf{x} \left(k \right)$ in the above described list. For example, if the second best matching unit is the sixth closest neighbor of the first best matching unit for some test sample, then we accumulate 6 to the computation of the MTR. If several units have the same topological distance to the first best matching unit i.e. there is a tie among them, then their average rank is used for MTR computation in case that one of them is the second best matching unit for some test sample. Please note that MTR does not make sense for competitive learning because no topology is defined, so it is not reported for this model in the presented results.

A statistical significance study has been carried out for all the quantitative performance measures. Once the map parameters for a proposal, network size and dataset are obtained by the procedure detailed in Subsection 4.1, 100 runs of hold-out cross-validation are executed, each with randomly split training set (90% of the samples) and test set (10% of the samples). The reported quantitative values are the mean and standard deviation computed over the 100 runs corresponding to the best performing map size for each competing method. After that, the nonparametric Friedman test with the corresponding post-hoc Dunn test are used to determine whether the difference of the best competing method with respect to all the others is statistically significant. These tests are robust for multi-way comparisons [24]. A 95% confidence level has been chosen in all cases.

The detailed results of the three performance measures for the machine data and image datasets are listed in Tabs. III to XI. The overall result of these exper-

	Competitive	1D SOM	2D SOM	3D SOM
BalanceScale	1.00 (0.00)	**0.91 (0.03)***	1.02 (0.05)	1.16 (0.07)
BrCanWis	7.71 (1.48)	**7.69 (1.52)**	9.71 (1.89)	11.42 (2.18)
Cloud	**551 (349)***	996 (1144)	2310 (2795)	3632 (4264)
Contraceptive	**2.06 (0.12)***	2.37 (0.15)	2.95 (0.27)	3.20 (0.31)
Dermatology	13.69 (1.21)	**11.69 (1.23)***	12.88 (1.80)	13.37 (1.95)
Glass	0.69 (0.56)	**0.64 (0.56)***	0.82 (0.63)	1.00 (0.72)
Haberman	**6.57 (3.35)**	**6.57 (3.52)**	8.55 (5.76)	10.3 (7.38)
HayesRoth	0.45 (0.21)	**0.38 (0.19)***	0.45 (0.21)	0.56 (0.22)
Liver	183 (71.9)	**166 (61.7)***	222 (106)	254 (146)
Pima	**350 (211)***	380 (273)	617 (594)	831 (747)
Vowel	**0.21 (0.02)***	0.33 (0.03)	0.702 (0.052)	0.92 (0.06)
Wine	249 (291)	**186 (293)**	271.9 (598.9)	372 (781)
Yeast	**.008 (.001)***	.011 (.001)	.015 (.002)	.017 (.003)

Tab. III *Mean Squared Error results for the small machine learning datasets experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

	Competitive	1D SOM	2D SOM	3D SOM
BalanceScale	0.84 (0.07)	**0.87 (0.05)***	0.78 (0.08)	0.70 (0.09)
BrCanWis	0.89 (0.05)	**0.90 (0.04)**	0.77 (0.07)	0.65 (0.08)
Cloud	**0.78 (0.03)***	0.73 (0.04)	0.59 (0.04)	0.53 (0.05)
Contraceptive	**0.80 (0.04)***	0.76 (0.05)	0.59 (0.06)	0.58 (0.05)
Dermatology	0.91 (0.06)	**0.94 (0.05)***	0.87 (0.07)	0.85 (0.06)
Glass	**0.96 (0.05)**	**0.96 (0.05)**	0.91 (0.07)	0.87 (0.10)
Haberman	0.94 (0.05)	**0.95 (0.05)**	0.93 (0.064)	0.90 (0.07)
HayesRoth	0.96 (0.06)	**0.98 (0.04)**	0.97 (0.06)	0.94 (0.074)
Liver	0.93 (0.06)	**0.94 (0.05)**	0.89 (0.07)	0.86 (0.08)
Pima	**0.89 (0.04)**	0.88 (0.04)	0.73 (0.07)	0.65 (0.08)
Vowel	**0.94 (0.02)***	0.91 (0.03)	0.71 (0.05)	0.62 (0.06)
Wine	**0.98 (0.03)**	**0.98 (0.03)**	**0.98 (0.03)**	0.97 (0.04)
Yeast	**0.78 (0.04)***	0.73 (0.05)	0.53 (0.06)	0.46 (0.07)

Tab. IV *Mean Silhouette Value results for the small machine learning datasets experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

iments is that the 1D SOM outperforms the other two SOMs in most cases with respect to MSE, MSV and MTR. An exception to this is the MSV for the image experiments, where 2D and 3D maps are better. It must be highlighted that the competitive learning networks are the best in terms of MSE, with very good MSV results, and that 1D SOMs are the second best in terms of MSE. This agrees with the theoretical results developed in Section 3, where it is highlighted that competitive learning minimizes the MSE, while 1D SOM minimizes an energy function which is closer to MSE than the energy functions of 2D SOMs and 3D SOMs. Consequently, the experiments indicate that 1D SOMs are the self-organizing maps of choice whenever vector quantization performance is the most important goal.

	1D SOM	2D SOM	3D SOM
BalanceScale	8.917 (1.952)	8.214 (1.837)	**7.562 (1.818)**
BrCanWis	**4.489 (1.101)**	4.522 (0.904)	5.118 (0.865)
Cloud	**2.658 (0.560)***	3.985 (0.427)	7.867 (1.035)
Contraceptive	**3.916 (0.519)***	4.943 (0.528)	6.673 (0.484)
Dermatology	**3.431 (0.969)**	3.576 (0.406)	6.085 (0.815)
Glass	4.580 (1.636)	**4.185 (1.466)**	4.742 (1.387)
Haberman	6.030 (2.033)	**3.551 (0.876)***	5.146 (0.882)
HayesRoth	**4.120 (2.934)**	5.003 (3.042)	5.181 (2.536)
Liver	6.893 (1.746)	**4.268 (1.143)***	5.620 (0.970)
Pima	**3.606 (0.662)***	4.711 (0.668)	7.290 (1.278)
Vowel	6.068 (1.282)	**5.559 (1.090)**	6.058 (1.112)
Wine	**1.751 (0.177)***	11.907 (3.087)	16.480 (3.710)
Yeast	7.951 (1.090)	**5.543 (0.928)**	6.075 (0.932)

Tab. V *Mean Tied Rank results for the small machine learning datasets experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

	Competitive	1D SOM	2D SOM	3D SOM
CorelColorHistogram	**0.67 (0.00)**	**0.67 (0.00)**	0.63 (0.01)	0.64 (0.00)
CorelColorMoments	**0.67 (0.00)**	**0.67 (0.00)**	0.64 (0.00)	0.63 (0.00)
CorelCoocTexture	**0.67 (0.00)**	**0.67 (0.00)**	0.65 (0.00)	0.62 (0.00)
CorelLayoutHistogram	**0.67 (0.00)**	**0.67 (0.00)**	0.63 (0.01)	0.64 (0.00)
CoverType	**0.33 (0.00)***	0.32 (0.00)	0.31 (0.00)	0.30 (0.00)
Letter	**0.26 (0.00)***	0.20 (0.01)	0.22 (0.01)	0.17 (0.01)
MiniBooNE	**0.26 (0.01)***	0.19 (0.00)	0.14 (0.00)	0.12 (0.02)
SkinSegmentation	**0.63 (0.01)***	0.49 (0.01)	0.49 (0.02)	0.44 (0.02)

Tab. VI *Mean Silhouette Value for the large machine learning datasets experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

	1D SOM	2D SOM	3D SOM
CorelColorHistogram	**1.478 (0.001)***	17.460 (1.026)	17.419 (0.984)
CorelColorMoments	**1.478 (0.001)***	17.139 (1.130)	18.199 (0.788)
CorelCoocTexture	**1.478 (0.001)***	16.792 (1.036)	18.569 (0.925)
CorelLayoutHistogram	**1.477 (0.001)***	17.599 (1.023)	18.927 (0.905)
CoverType	8.663 (0.719)	**4.449 (0.238)***	9.785 (0.619)
Letter	**8.746 (0.803)***	13.647 (1.089)	15.545 (1.342)
MiniBooNE	5.655 (0.596)	**4.279 (0.392)***	7.006 (0.841)
SkinSegmentation	**3.646 (0.303)***	4.047 (0.269)	7.914 (0.907)

Tab. VII *Mean Tied Rank results for the large machine learning datasets experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

	Competitive	1D SOM	2D SOM	3D SOM
Baboon	0.38 (0.00)	0.36 (0.00)	0.38 (0.00)	**0.39 (0.00)***
F16	0.39 (0.00)	**0.41 (0.00)***	0.37 (0.00)	0.36 (0.00)
House	**0.38 (0.01)**	0.28 (0.00)	**0.38 (0.01)**	0.35 (0.01)
Lake	**0.38 (0.00)***	0.33 (0.00)	0.37 (0.00)	0.36 (0.00)
Lena	**0.40 (0.00)**	0.38 (0.00)	**0.40 (0.00)**	0.38 (0.00)
Peppers	**0.41 (0.00)**	0.39 (0.00)	**0.41 (0.00)**	**0.41 (0.01)**

Tab. VIII *Mean Silhouette Value results for the image segmentation experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

	1D SOM	2D SOM	3D SOM
Baboon	7.270 (0.597)	**4.554 (0.337)***	7.918 (0.609)
F16	**3.621 (0.092)***	6.696 (0.250)	13.000 (0.689)
House	**2.692 (0.095)***	4.284 (0.480)	6.126 (0.630)
Lake	**4.712 (0.196)***	5.579 (0.339)	8.420 (0.709)
Lena	4.951 (0.172)	**3.621 (0.241)***	9.907 (0.535)
Peppers	4.936 (0.598)	**3.659 (0.174)***	7.395 (0.613)

Tab. IX *Mean Tied Rank results for the image segmentation experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

	Competitive	1D SOM	2D SOM	3D SOM
CorelColorHistogram	**996 (40)**	999 (79)	5.24×10^4 (9673)	1.96×10^5 (2.10×10^4)
CorelColorMoments	**1007 (38)**	1018 (74)	5.02×10^4 (9669)	1.90×10^5 (2.24×10^4)
CorelCoocTexture	**1016 (36)**	1045 (72)	5.83×10^4 (1.28×10^4)	1.88×10^5 (1.95×10^4)
CorelLayoutHistogram	**963 (32)**	986 (70)	5.53×10^4 (9463)	2.14×10^5 (2.21×10^4)
CoverType	$\mathbf{5.26\times10^4}$ **(378)***	8.10×10^4 (527)	9.61×10^4 (1105)	5.32×10^4 (345)
Letter	**8.92 (0.12)***	13.40 (0.18)	18.85 (0.36)	21.78 (0.45)
MiniBooNE	$\mathbf{1.78\times10^9}$ ($\mathbf{5.69\times10^9}$)	$\mathbf{1.78\times10^9}$ ($\mathbf{5.69\times10^9}$)	$\mathbf{1.78\times10^9}$ ($\mathbf{5.69\times10^9}$)	$\mathbf{1.78\times10^9}$ ($\mathbf{5.69\times10^9}$)
SkinSegmentation	**30.6 (3.0)***	83.9 (5.1)	199 (10)	339 (21)

Tab. X *Mean Squared Error for the large machine learning datasets experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

	Competitive	1D SOM	2D SOM	3D SOM
Baboon	$\mathbf{8.366\times10^{-4}}$ ($\mathbf{7.241\times10^{-6}}$)*	9.676×10^{-4} (1.512×10^{-5})	1.437×10^{-3} (9.777×10^{-6})	8.455×10^{-4} (5.605×10^{-6})
F16	$\mathbf{1.962\times10^{-4}}$ ($\mathbf{8.500\times10^{-6}}$)*	3.068×10^{-4} (2.057×10^{-5})	1.020×10^{-3} (1.535×10^{-4})	1.498×10^{-3} (1.423×10^{-4})
House	$\mathbf{2.047\times10^{-4}}$ ($\mathbf{7.017\times10^{-6}}$)*	3.619×10^{-4} (1.475×10^{-5})	5.100×10^{-4} (2.128×10^{-5})	3.463×10^{-4} (7.584×10^{-6})
Lake	$\mathbf{5.219\times10^{-4}}$ ($\mathbf{9.546\times10^{-6}}$)*	1.037×10^{-3} (3.283×10^{-5})	1.171×10^{-3} (4.412×10^{-5})	1.582×10^{-3} (6.783×10^{-5})
Lena	$\mathbf{2.713\times10^{-4}}$ ($\mathbf{3.374\times10^{-6}}$)*	4.842×10^{-4} (8.363×10^{-6})	6.525×10^{-4} (2.691×10^{-5})	8.675×10^{-4} (3.728×10^{-5})
Peppers	$\mathbf{4.841\times10^{-4}}$ ($\mathbf{8.254\times10^{-6}}$)*	1.081×10^{-3} (5.919×10^{-5})	1.034×10^{-3} (5.816×10^{-5})	2.157×10^{-3} (1.022×10^{-4})

Tab. XI *Mean Squared Error results for the image segmentation experiments. Best results are marked in* ***bold****. An asterisk indicates that the difference among the best method and all the others is statistically significant.*

5. Discussion

We start by contrasting the practical aspects of neuron density (Subsections 4.2) with the theoretical claims in Subsections 3.1, 3.3 and 3.4:

- In Figs. 3 to 5, as we navigate towards the right-hand side of the figure, the neuron density grows near the expectation of the input density. Thus this behavior coincides with the foreseen in Subsection 3.1: the larger the lattice dimensionality the stronger the influence of the expected value of the samples.

- With reference to Subsection 3.3, in Figs. 3 to 5 we may observe that the lattice matching the input dimensionality, i.e., 2D SOM, has the neuron density which is less blurred. So this topology achieves the smallest MSE values on the average, confirming the conclusions of that Subsection.

- Lastly, studying Figs. 3 to 7, we perceive that the lower the dimension the lower the number of zones of high density and consequently the more bonded the SOM is to local optimums. This was predicted in Subsection 3.4.

Next, key findings related to applications are discussed:

- The energy function of the 2D SOM departs from the optimization of the MSE further away than that of the 1D SOM (Subsection 3.1). This suggests that 1D SOMs are more suited to applications where vector quantization is important. On the other hand, 3D SOMs have the energy function which is the farthest from the MSE, so it is more natural to employ it for applications where learning a three dimensional structure is more relevant. The behavior of the SOM near a local minimum of the MSE also favors 1D SOMs for these applications (Subsection 3.4).

- The intrinsic dimensionality of the input dataset influences which topology performs best (Subsection 3.3). The lattice with the matching dimensionality should be chosen if possible. Moreover, the specific structure of the input dataset deeply influences the unfolding of the SOM, and each lattice dimensionality exhibits a completely different behavior (Subsections 3.2 and 4.2). Emergence of complex patterns in the neuron density function of the 2D SOM has been discovered (Fig. 4).

- Experiments with real data (Subsection 4.3) indicate that 1D topologies are the best SOMs overall, not only with respect to MSE but also to topological map quality. On the other hand, the competitive learning networks attain even better values of MSE. These results agree with the theory developed in Section 3.

From the above considerations it can be said that 1D and 3D lattice topologies are heavily underutilized, since they clearly outperform the standard 2D in many respects.

6. Conclusions

Three alternative grid topologies for self-organizing maps have been examined. A theoretical study of them has been carried out from several points of view. Experiments have been carried out over synthetic and real data to compare them. Several quantitative performance measures have been chosen to this end, and the statistical significance of the results has been computed. The results and the further discussion indicate that the 1D and 3D topologies are well suited to many datasets. This indicates that there is room to improve SOM-based systems by employing these relatively uncommon topologies.

Acknowledgement

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grant TIN2014-53465-R, project name Video surveillance by active search of anomalous events. It is also partially supported by the Autonomous Government of Andalusia (Spain) under project P12-TIC-657, project name Self-organizing systems and robust estimators for video surveillance. All of them include funds from the European Regional Development Fund (ERDF). The second author is supported by 'Ramón y Cajal' grant RYC-2010-05663 from the Ministry of Science and Innovation of Spain. The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga.

References

[1] ASUNCION A., NEWMAN D.J. UCI machine learning repository, 2007. URL http://www.ics.uci.edu/mlearn/MLRepository.html.

[2] KAEVER A., LINGNER T., FEUSSNER K., GÖBEL C., FEUSSNER I., MEINICKE P. MarVis: A tool for clustering and visualization of metabolic biomarkers. *BMC Bioinformatics*, 10, 2009.

[3] MEYER-BAESE A., WISMUELLER A., LANGE O. Comparison of two exploratory data analysis methods for fMRI: Unsupervised clustering versus independent component analysis. *IEEE Transactions on Information Technology in Biomedicine*, 8(3), pp. 387–398, 2004.

[4] HSU A.L., HALGAMUGE S.K. Enhancement of topology preservation and hierarchical dynamic self-organising maps for data visualisation. *International Journal of Approximate Reasoning*, 32(2–3), pp. 259–279, 2003.

[5] WEBER A. USC-SIPI Image Database, 2010. URL http://sipi.usc.edu/database/.

[6] HUNEITI A.M. Interpreting web usage patterns generated using a hybrid SOM-based clustering technique. *International Review on Computers and Software*, 7(3), pp. 1078–1088, 2012.

[7] AZCARRAGA A., MANALILI S. Design of a structured 3D SOM as a music archive. In Jorma Laaksonen and Timo Honkela, editors, *Advances in Self-Organizing Maps*, volume 6731 of *Lecture Notes in Computer Science*, pp. 188–197. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-21565-0.

[8] BARUQUE B., CORCHADO E. A weighted voting summarization of SOM ensembles. *Data Mining and Knowledge Discovery*, 21(3), pp. 398–426, 2010.

[9] CURRY B., MORGAN P.H. Evaluating kohonen's learning rule: An approach through genetic algorithms. *European Journal of Operational Research*, 154(1), pp. 191–205, 2004.

[10] MAHALAKSHMI B., DURAISWAMY K. Efficient categorization of web documents using enhanced self-organizing map. *European Journal of Scientific Research*, 77(2), pp. 231–239, 2012.

[11] WIJAYASEKARA D., LINDA O., MANIC M. CAVE-SOM: Immersive visual data mining using 3d self-organizing maps. In: *The 2011 International Joint Conference on Neural Networks*, pp. 2471–2478, 2011.

[12] KAYE D., IVRISSIMTZIS I. Implicit surface reconstruction and feature detection with a learning algorithm. In: John Collomosse and Ian Grimstead, editors, *Theory and Practice of Computer Graphics*, pp. 127–130. European Association for Computer Graphics, 2010.

[13] BEATON D., VALOVA I., MACLEAN D. CQoCO: A measure for comparative quality of coverage and organization for self-organizing maps. *Neurocomputing*, 73(10-12), pp. 2147–2159, 2010.

[14] CORCHADO E., BARUQUE B. WeVoS-ViSOM: An ensemble summarization algorithm for enhanced data visualization. *Neurocomputing*, 75(1), pp. 171–184, 2012.

[15] LÓPEZ-RUBIO E., DÍ AZ R.A. Grid topologies for the self-organizing map. *Neural Networks*, 56C, pp. 35–48, 2014.

[16] LÓPEZ-RUBIO E., LUQUE-BAENA R.M., DOMÍ NGUEZ E. Foreground detection in video sequences with probabilistic self-organizing maps. *International Journal of Neural Systems*, 21(3), pp. 225–246, 2011.

[17] BOUDJEMAI F., ENBERG P.B., POSTAIRE J.-G. Dynamic adaptation and subdivision in 3D-SOM application to surface reconstruction. In: *17th IEEE International Conference on Tools with Artificial Intelligence*, pp. 6–430, 2005.

[18] CHICCO G., NAPOLI R., PIGLIONE F. Comparisons among clustering techniques for electricity customer classification. *IEEE Transactions on Power Systems*, 21(2), pp. 933–940, 2006.

[19] YIN H. The self-organizing maps: Background, theories, extensions and applications. *Studies in Computational Intelligence*, 115, pp. 715–762, 2008.

[20] KUSHNER H.J., YIN G.G. *Stochastic approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, NY, USA, 2003.

[21] BARHAK J., FISCHER A. Adaptive reconstruction of freeform objects with 3D SOM neural network grids. In: *Proceedings of the Ninth Pacific Conference on Computer Graphics and Applications*, pp. 97–105, 2001.

[22] FORT J.C. SOM's mathematics. *Neural Networks*, 19(6), pp. 812–816, 2006.

[23] FLANAGAN J.A. Self-organized criticality and the self-organizing map. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 63(3 II), pp. 361301–361306, 2001.

[24] DEMŠAR J. Statistical comparisons of classifiers over multiple data sets. *Jounal of Machine Learning Research*, 7(1):1–30, 2006. ISSN 1532-4435.

[25] NELDER J.A., MEAD R. A simplex method for function minimization. *Computer Journal*, 7, pp. 308–313, 1965.

[26] GORRICHA J., LOBO V. Improvements on the visualization of clusters in geo-referenced data using self-organizing maps. *Computers and Geosciences*, 43, pp. 177–186, 2012. ISSN 0098-3004.

[27] TASDEMIR K. Vector quantization based approximate spectral clustering of large datasets. *Pattern Recognition*, 45(8), pp. 3034–3044, 2012.

[28] COTTRELLA M., FORTB J.C., PAGÉSC G. Theoretical aspects of the SOM algorithm. *Neurocomputing*, 21(1-3), pp. 119–138, 1998.

[29] HÜSER M., ZHANG J. Visual programming by demonstration of grasping skills in the context of a mobile service robot using 1D-topology based self-organizing-maps. *Robotics and Autonomous Systems*, 60(3), pp. 463–472, 2012. ISSN 0921-8890.

[30] MERKOW M., DELISLE R.K. Improving the performance of self-organizing maps via growing representations. *Journal of Chemical Information and Modeling*, 47(5), pp. 1797–1807, 2007.

[31] VASSILAS N. Self-organization of the batch Kohonen network under quantization effects. *International Journal of Computer Mathematics*, 88(17), pp. 3586–3612, 2011.

[32] DLUGOSZ R., TALASKA T., PEDRYCZ W., WOJTYNA R. Realization of the conscience mechanism in CMOS implementation of winner-takes-all self-organizing neural networks. *IEEE Transactions on Neural Networks*, 21(6), pp. 961–971, 2010.

[33] XU R., XU J., WUNSCH D.C. A comparison study of validity indices on swarm-intelligence-based clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4), pp. 1243–1256, 2012.

[34] DATTA S., DATTA S. Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. *BMC Bioinformatics*, 7(Suppl 4), pp. S17, 2006.

[35] OKAJIMA S., OKADA Y. Treecube+3D-ViSOM: Combinational visualization tool for browsing 3D multimedia data. In: *11th International Conference on Information Visualization*, pp. 40–45, 2007.

[36] KANNAN S.R., RAMATHILAGAM S., CHUNG P.C. Effective fuzzy c-means clustering algorithms for data clustering problems. *Expert Systems with Applications*, 39(7), pp. 6292–6300, 2012.

[37] ABEEL T., SAEYS Y., ROUZÉ P., VAN DE PEER Y. ProSOM: Core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics*, 24(13), pp. i24–i31, 2008.

[38] KOHONEN T. Comparison of SOM point densities based on different criteria. *Neural Computation*, 11(8), pp. 2081–2095, 1999.

[39] KOHONEN T. The self-organizing map. *Proceedings of the IEEE*, 78(9), pp. 1464–1480, 1990.

[40] KOHONEN T. *Self-Organizing Maps*. Springer, 3nd edition, 2001.

[41] KOHONEN T. Essentials of the self-organizing map. *Neural Networks*, 37, pp. 52–65, 2013. ISSN 0893-6080.

[42] SEIFFERT U., MICHAELIS B. Growing 3D-SOMs with 2D-input layer as a classification tool in a motion detection system. *International Journal of Neural Systems*, 8(1), pp. 81–89, 1997.

[43] MORARIU V.I., RAYKAR VIKAS C., YANG CH. FIGTree library, 2007. URL: http://www.umiacs.umd.edu/~morariu/figtree/.

[44] MORARIU VLAD I., SRINIVASAN B.V., RAYKAR VIKAS C., DURAISWAMI R., DAVIS L.S. Automatic online tuning for fast Gaussian summation. *Advances in Neural Information Processing Systems*, 21, pp. 1113–1120, 2008.

[45] XIONG Y., WALLACH R., FURMAN A. Modeling multidimensional flow in wettable and water-repellent soils using artificial neural networks. *Journal of Hydrology*, 410(1-2), pp. 92–104, 2011.

[46] YANG Y., TAN W., LI T., RUAN D. Consensus clustering based on constrained self-organizing map and improved cop-kmeans ensemble in intelligent decision support systems. *Knowledge-Based Systems*, 32, pp. 101–115, 2012.

[47] XIAO YI, LEUNG CHI-SING, LAM PING-MAN, HO TZE-YUI. Self-organizing map-based color palette for high-dynamic range texture compression. *Neural Computing and Applications*, 21(4), pp. 639–647, 2012. ISSN 0941-0643.