Online data visualization using the neural gas network

Pablo A. Estévez*, Cristián J. Figueroa

Department of Electrical Engineering, University of Chile, Casilla 412-3, Santiago, Chile

Abstract

A high-quality distance preserving output representation is provided to the neural gas (NG) network. The nonlinear mapping is determined concurrently along with the codebook vectors. The adaptation rule for codebook positions in the projection space minimizes a cost function that favors the trustworthy preservation of the local topology. The proposed visualization method, called OVI-NG, is an enhancement over curvilinear component analysis (CCA). The results show that the mapping quality obtained with OVI-NG outperforms the original CCA, in terms of the trustworthiness, continuity, topographic function and topology preservation measures.

Keywords: Neural gas; Data projection; Data visualization; Topology preservation; Distance preservation

1. Introduction

The self-organizing feature map (SOM) (Kohonen, 1995) has been widely used for vector quantization (VQ) and for data projection and visualization. The VQ techniques encode a manifold of data by using a finite set of reference or "codebook" vectors. The SOM performs VQ under the constraint of a predefined neighborhood between neurons in a discrete output grid. In this way the SOM yields a topographic mapping from the manifold to the output grid. In the conventional SOM the distances between codebook vectors are not directly represented in the map. Post-processing techniques such as Cluster Connections (Merkl & Rauber, 1997), P-matrix (Ultsch, 2003) or U-matrix (Ultsch & Siemon, 1990) allow us to incorporate the distance information in the conventional output display by using coloring schemes. Other extensions to the SOM are Adaptive Coordinates (Merkl & Rauber, 1997) and Double SOM (Su & Chang, 2001), which allow us to visualize the original structure of the data in a low-dimensional output space without using post-processing techniques. Both models use a heuristic updating rule to move and group the output nodes in a continuous output space, but they do not preserve the intra-cluster and inter-cluster distances as well as Sammon's mapping (Sammon, 1969). Another SOM extension is DIPOL-SOM (König & Michel, 2003), which computes a distance preserving projection, and moves nodes in an additional projection layer by using a heuristic online adaptation rule.

A visualization induced SOM (ViSOM) has been proposed to preserve the inter-neuron distances in the map (Yin, 2002a, 2002b). The ViSOM constrains the lateral contraction force between neurons in the SOM, allowing us to preserve the inter-point distances on the input data on the map, along with the topology. However, the ViSOM uses the same fixed grid structure of neurons as SOM, and imposes a uniform distribution of the codebook vectors in the input space. As a consequence, ViSOM requires a large number of codebooks to get an adequate quantization error, which entails a heavy computational load. The resolution and the computational cost can be enhanced by interpolating a trained map or incorporating local linear projections (Yin, 2003).

Another enhancement to the SOM is curvilinear component analysis (CCA) (Demartines & Hérault, 1997). Firstly, CCA performs VQ of the data manifold in input space using SOM. Secondly, CCA makes a nonlinear projection of the quantizing vectors. The projection part of CCA is similar to multi-dimensional scaling (MDS) (Cox & Cox, 2001) or Sammon's mapping (NLM) (Sammon, 1969), since it minimizes a cost function based on the inter-point distances. However, the computational complexity of CCA is $O(N)$, while MDS and NLM are $O(N^2)$. Another difference is that the cost function of CCA allows us to unfold strongly nonlinear or closed

* Corresponding author. Tel.: +56 2 9784207; fax: +56 2 6720162.
E-mail addresses: pestevez@ing.uchile.cl (P.A. Estévez), cristian.figueroa@sonda.com (C.J. Figueroa).

structures. In CCA, the output is not a fixed grid but a continuous space that is able to take the shape of the data manifold. The codebook vectors are projected as codebook positions in output space, which are updated by a special adaptation rule. An enhanced version of CCA incorporates curvilinear distances instead of Euclidean distances in the input space (Lee, Lendasse, Donckers, & Verleysen, 2000; Lee, Lendasse, & Verleysen, 2004).

The neural gas (NG) is another well-known self-organizing neural network (Martinetz & Schulten, 1991). The main difference with SOM is that NG does not define an output space. The SOM is able to obtain good VQ results only if the topology of the output grid matches the topology of the data manifold, which is usually unknown. As a consequence, NG can outperform SOM when quantizing topologically arbitrary structured manifolds. Instead of a neighborhood function in the output grid, the NG utilizes a neighborhood ranking of the codebook vectors within the input space. In addition, the NG adaptation rule for codebook vectors obeys a stochastic gradient descent of a global cost function, while no cost function exists for the SOM adaptation rule (Martinetz, Berkovich, & Schulten, 1993). The NG algorithm can be combined with the competitive Hebbian rule (CHR), for forming connections among neighboring units (Martinetz & Schulten, 1994). The procedure forms induced Delaunay triangulations of the distribution of codebook vectors.

The lack of an output space has limited the application of NG to data projection and visualization. After learning, the codebook vectors can be projected onto a two-dimensional space using MDS or Sammon's mapping. Alternatively, the authors proposed in Figueroa and Estévez (2004) an online visualization method called TOPNG, which updates simultaneously the codebook vectors in input space and their corresponding vector positions in a continuous output space. The TOPNG adds to NG a simple heuristic adaptation rule for codebook positions. When TOPNG is combined with particle swarm optimization, the Sammon error is further minimized (Figueroa, Estévez, & Hernández, 2005). In a different approach, a cross-entropy embedding of high-dimensional data using the neural gas model has been proposed (Estévez, Figueroa, & Saito, 2005a, 2005b). This method allows us to project simultaneously the input data and the codebook vectors onto a projection plane, preserving the relationship defined by the NG neighborhood ranking function.

As pointed out in Venna and Kaski (2005), every projection method necessarily makes a tradeoff between trustworthiness and continuity. A projection is said to be trustworthy if the visualized proximities hold in the original space, and it is continuous if it visualizes all the proximities of the original data. Of these two types of measures, it has been argued that trustworthiness is more important, since it guarantees that at least a portion of the similarities will be perceived correctly (Kaski et al., 2003). It has been shown that SOM and CCA projections have a high trustworthiness, with a clear difference over other methods such as Isomap (Tenenbaum, de Silva, & Langford, 2000) and Local Linear Embedding (Roweis & Saul, 2000). Another way of assessing the mapping quality is by using the topology preservation measure defined in König (2000), which takes rank order into account. A more precise definition of topology preservation was given by Villmann, Der, Herrmann, and Martinetz (1997), along with a tool for measuring it — the topographic function. Although this measure was defined for SOM with rectangular lattices, an extension to the case of more general lattices was proposed too.

In this paper we propose a distance preserving nonlinear mapping of the codebook vectors (prototypes) obtained with the NG algorithm. The codebook positions (projections) are adjusted in a continuous output space by using an adaptation rule that minimizes a cost function that favors the local distance preservation. The proposed strategy is online, i.e. the codebook positions are updated simultaneously with the codebook vectors. The performance of the proposed mapping method is compared with SOM-CCA and NG-CCA, on several data sets, in terms of the trustworthiness, continuity, topographic function and topology preserving measures. Preliminary results showed that the proposed method outperformed DIPOL-SOM, TOPNG, SOM/NLM and NG/NLM in terms of the topology preservation measure (Estévez & Figueroa, 2005).

2. The online visualization neural gas algorithm

The proposed visualization method is called OVI-NG (Online VIsualization Neural Gas).

2.1. Problem setting

Let $\{x_i : 1 \leq i \leq M\}$ and $\{w_j : 1 \leq j \leq N\}$ be D-dimensional input and codebook vectors, respectively. For a given set of input vectors, our problem is to adjust simultaneously the codebook vectors in the input space and their respective codebook positions $z_j (j = 1, \ldots, N)$ in an A-dimensional continuous output space, with $A \ll D$.

In order to obtain a distance preserving mapping, a cost function is defined, which depends on the difference between the inter-point distances in both the input and the output spaces. Let $d_{j,k}$ and $D_{j,k}$ be the Euclidean distances defined in the input and the output spaces, respectively:

$$d_{j,k} = \|\mathbf{w}_j - \mathbf{w}_k\| = \sqrt{\sum_{d=1}^{D} (w_{j,d} - w_{k,d})^2}, \tag{1}$$

$$D_{j,k} = \|\mathbf{z}_j - \mathbf{z}_k\| = \sqrt{\sum_{\ell=1}^{A} (z_{j,\ell} - z_{k,\ell})^2}. \tag{2}$$

In the neural gas model, the neighborhood ranking function of the codebook vectors $\mathbf{w}_j$, for $j = 1, \ldots, N$, with respect to a given input vector x_i, is defined as follows:

$$h_\lambda(x_i, w_j) = \mathrm{e}^{-\frac{r(x_i(t), w_j(t))}{\lambda(t)}}, \tag{3}$$

where $r_{ij} = r(x_i, w_j) \in \{0, 1, \ldots, N-1\}$ denotes the rank of the jth codebook vector, and the parameter $\lambda(t)$ controls the

width of the neighborhood function,

$$\lambda(t) = \lambda_0 \left(\frac{\lambda_f}{\lambda_0}\right)^{\left(\frac{t}{t_{\max}}\right)}, \tag{4}$$

where $t_{\max}$ is the maximum number of adaptation steps.

Likewise, we introduce a ranking of the codebook positions in output space $\mathbf{z}_j$, $j = 1, \ldots, N$, with respect to a given output vector y_i. The term $s_{ij} = s(y_i, z_j) \in \{0, 1, \ldots, N-1\}$ denotes the rank of the jth codebook position. Here $r = 0$ and $s = 0$ are associated with the nearest codebook vector and the nearest codebook position, respectively.

2.2. Codebook position adaptation rule

A global cost function similar to the one used by CCA (Demartines & Hérault, 1997) is introduced:

$$E = \frac{1}{2}\sum_{j=1}^{N}\sum_{k\neq j}(D_{j,k} - d_{j,k})^2 F(s_{j,k}) = \frac{1}{2}\sum_{j=1}^{N}\sum_{k\neq j} E_{j,k}, \tag{5}$$

where the function F is defined as

$$F(f) = \mathrm{e}^{-\left(\frac{f}{\sigma(t)}\right)} \tag{6}$$

and $\sigma(t)$ is the width of the neighborhood that decreases with the number of iterations in the same way as Eq. (4). The function $F(f)$ is a bounded and monotonically decreasing function, in order to favor local topology preservation.

Eq. (5) could be minimized with respect to z_j by using stochastic gradient descent. In that case, a codebook position would be adjusted according to the sum of every other codebook influence. For each adaptation cycle of all N units, $N \times (N-1)/2$ distances should be computed, in both input space and output space. Therefore, the complexity of such an algorithm is $O(N^2)$.

Instead, we use a simplified version of gradient descent, where the codebook position associated with the winner unit, $z_j^*(t)$, is fixed, and the $N-1$ remaining positions are moved towards the winner's position, disregarding any interaction between them. In this case, the ranking in output space s_{j,j^*} takes values in the range $\{1, \ldots, N-1\}$, where $s = 1$ corresponds to the nearest codebook position with respect to the winner's position. Using this method, the cost function will decrease on average, as shown in Demartines and Hérault (1997). The complexity of the proposed visualization algorithm is $O(N \log N)$, due to the determination of the ranking in output space. Since the computational complexity of NG is $O(N \log N)$ (Martinetz et al., 1993), the combined algorithm of NG as VQ and our online visualization strategy also has complexity $O(N \log N)$.

This approach is similar to CCA, but CCA is an offline method since it performs the nonlinear mapping after the codebook vectors have been learned by SOM. Our approach is online and it uses NG as a vector quantizer instead of SOM. Another difference is that, in our approach, the weighting function F in (6) is an exponential function of the ranking of position vectors in the output space, while in CCA the weighting function F is a step function of the inter-point Euclidean distances in output space. In order to minimize (5), in OVI-NG the derivative of the local objective function E_{j,j^*} with respect to position vector z_j, maintaining z_{j^*} fixed, is calculated by utilizing the chain rule as follows:

$$\frac{\partial E_{j,j^*}}{\partial z_j} = F(s_{j,j^*})\frac{(D_{j,j^*} - d_{j,j^*})}{D_{j,j^*}}(z_j - z_{j^*}), \tag{7}$$

where the quantized function F is considered constant with respect to the variation of the codebook positions $z_j(t)$, independent of the choice of F and the distance measurement in output space. Consequently, the updating rule for the codebook positions is as follows:

$$z_j(t+1) = z_j(t) + \alpha(t)F(s_{j,j^*})\frac{(D_{j,j^*} - d_{j,j^*})}{D_{j,j^*}} \times (z_{j^*}(t) - z_j(t)), \tag{8}$$

where $\alpha(t)$ is the learning rate, which typically decreases with the number of iterations t, in the same form as Eq. (4).

2.3. OVI-NG learning algorithm

The learning algorithm combines NG for VQ, with CHR for forming connections between units, and an adaptation rule for codebook positions.

The initial topology of the network is a set of N neurons. Each neuron j has associated a D-dimensional codebook vector, w_j, and a two-dimensional codebook position, z_j ($j = 1, \ldots, N$). Let $\mathcal{C}_{in}$ be a connection set, which includes the connections between units in input space. Each connection $j^* - k^*$ has an age ($\mathrm{age}_{(j^*k^*)}$) that is defined as the number of adaptation steps without being refreshed since its creation. A connection is removed if its age exceeds its lifetime $T(t)$.

1. Initialize the codebook vectors, w_j, and the codebook positions, z_j, randomly. Set the connection set to the empty set: $\mathcal{C}_{in} = \emptyset$.
2. Present an input vector, $x_i(t)$, to the network ($i = 1, \ldots, M$) at iteration t.
3. Find the best matching unit (BMU), j^*, using:

   $$j^* = \underset{j=1\ldots N}{\operatorname{argmin}} \|x_i(t) - w_j(t)\|, \tag{9}$$

   and generate the ranking $r_{ij} = r(x_i(t), w_j(t)) \in \{0, 1, \ldots, N-1\}$ for each codebook vector $w_j(t)$ with respect to the input vector $x_i(t)$.
4. Update the codebook vectors:

   $$w_j(t+1) = w_j(t) + \epsilon(t)h_\lambda(t)(x_i(t) - w_j(t)), \tag{10}$$

   where

   $$h_\lambda(t) = h_\lambda(x_i, w_j) \tag{11}$$

   is the neighborhood ranking function defined in (3) and $\epsilon(t)$ is the learning rate, which depends on the number of adaptation steps, t, in the same form as (4).
5. If a connection between the BMU and the second closest unit does not exist already, create a connection between units

j^*-k^*: $\mathcal{C}_{in} = \mathcal{C}_{in} \cup \{w_j^*, w_k^*\}$, where the index of the second nearest unit is:

$$k^* = \underset{j \neq j^*}{\operatorname{argmin}} \|x_i(t) - w_j(t)\|. \tag{12}$$

Set the age of the connection $j^* - k^*$ to zero ("refresh" the connection if the connection exists):

$$\text{age}_{(j^*,k^*)} = 0. \tag{13}$$

6. Increment the age of all edges emanating from w_j^*:

$$\text{age}_{(j^*,\ell)} = \text{age}_{(j^*,\ell)} + 1, \forall \ell \in \mathcal{N}_{w_j^*}, \tag{14}$$

where $\mathcal{N}_{w_j^*}$ is the set of all direct topological neighbors of w_j^*. Remove those connections with an age exceeding the lifetime $T(t)$, which has the same dependency on time t as (4).

7. Generate the ranking in output space $s_{j^*j} = s(z_j^*(t), z_j(t)) \in \{1, \ldots, N-1\}$ for each codebook position $z_j(t)$ with respect to the codebook position associated with the winner unit $z_j^*(t)$, $j \neq j^*$.
8. Update the codebook positions:

$$z_j(t+1) = z_j(t) + \alpha(t) F(s_{j,j^*}) \times \frac{(D_{j,j^*} - d_{j,j^*})}{D_{j,j^*}} (z_{j^*}(t) - z_j(t)). \tag{15}$$

9. If $t < t_{\max}$, go back to step 2.

2.4. Mapping quality

The trustworthiness and continuity measures introduced in Kaski et al. (2003) and Venna and Kaski (2005) were used to compare the mapping quality of the proposed method with other CCA-like projection methods. A projection onto an output space is said to be trustworthy if the set of k nearest neighbors of a point in the map are also close by in the original space. Let $U_k(i)$ be the set of data samples that are in the neighborhood of the ith point in the map but not in the original space. The measure of trustworthiness of the visualization, M_1, is defined as:

$$M_1(k) = 1 - A(k) \sum_{i=1}^{N} \sum_{j \in U_k(i)} (r(w_i, w_j) - k), \tag{16}$$

where $A(k) = 2/(Nk \times (2N - 3k - 1))$, and $r(w_i, w_j)$ is the ranking in input space.

A projection onto an output space is said to be continuous if the set of k closest neighbors of a point in the original space are also close by in the output space. Let $V_k(i)$ be the set of data samples that are in the neighborhood of the ith point in the original space but not in the map. The measure of continuity of the visualization, M_2, is defined as:

$$M_2(k) = 1 - A(k) \sum_{i=1}^{N} \sum_{j \in V_k(i)} (s(z_i, z_j) - k), \tag{17}$$

where $s(z_i, z_j)$ is the ranking in output space.

In addition, the topology preservation measure q_m defined in König (2000) was considered as another performance measure. It is based on an assessment of rank order in the input and output spaces. The n nearest codebook vectors NN_{ji^w} ($i \in [1, n]$) of each codebook vector j ($j \in [1, N]$) and the n nearest codebook positions, NN_{ji^z}, of each codebook position j are computed. The global q_m measure is defined as:

$$q_m = \frac{1}{3nN} \sum_{j=1}^{N} \sum_{i=1}^{n} q_{m_{ji}}, \tag{18}$$

where

$$q_{m_{ji}} = \begin{cases} 3, & \text{if } NN_{ji^w} = NN_{ji^z} \\ 2, & \text{if } NN_{ji^w} = NN_{jl^z}, l \in [1, n], i \neq l \\ 1, & \text{if } NN_{ji^w} = NN_{jt^z}, t \in [n, k], n < k \\ 0, & \text{else.} \end{cases} \tag{19}$$

In our experiments, n was set to $n = 0.4 \times k$, and k was varied from 5 to a maximum value K. The range of the q_m measure is between 0 and 1, where $q_m = 0$ indicates poor neighborhood preservation between the input and output spaces, and $q_m = 1$ indicates a perfect neighborhood preservation.

We also use a representation of the mapping quality in terms of distance preservation, called "dz–dw" representation, which plots output distances versus input distances. This kind of representation was introduced in Demartines and Hérault (1997).

The topographic function introduced in Villmann et al. (1997) was also used to assess the mapping quality of the proposed method. Let M be a data manifold in input space, and A be a lattice of N neurons in output space. To each neuron $j \in A$, a codebook vector w_j is assigned. The map of M formed by A is defined by the map $\Psi_{M \to A}$ from M to A and the inverse mapping $\Psi_{A \to M}$ from A to M. The neighborhood preservation of the mappings $\Psi_{M \to A}$ and $\Psi_{A \to M}$ are denoted by $f_j(k)$ and $f_j(-k)$, respectively, where j is the index of the node in the map, and $k = 1, \ldots, N-1$. The topographic function is defined by,

$$\Phi_A^M(k) = \begin{cases} \dfrac{1}{N} \displaystyle\sum_{j \in A} f_j(k) & k < 0 \\ \Phi_A^M(1) + \Phi_A^M(-1) & k = 0 \\ \dfrac{1}{N} \displaystyle\sum_{j \in A} f_j(k) & k > 0 \end{cases} \tag{20}$$

where $\Phi_A^M(0) = 0$ if and only if the map is perfectly topology preserving. Calculating Φ_A^M requires us to determine Delaunay graphs in both M and A, as well as the connectivity graph in A that is generated by the competitive Hebbian rule. See Villmann et al. (1997) and Marsland, Shapiro, and Nehmzow (2002) for further details on the computation of the topographic function for general lattices.

2.5. Other CCA-like methods

For comparison purposes, four CCA-like strategies were considered. First, the original CCA method is here called

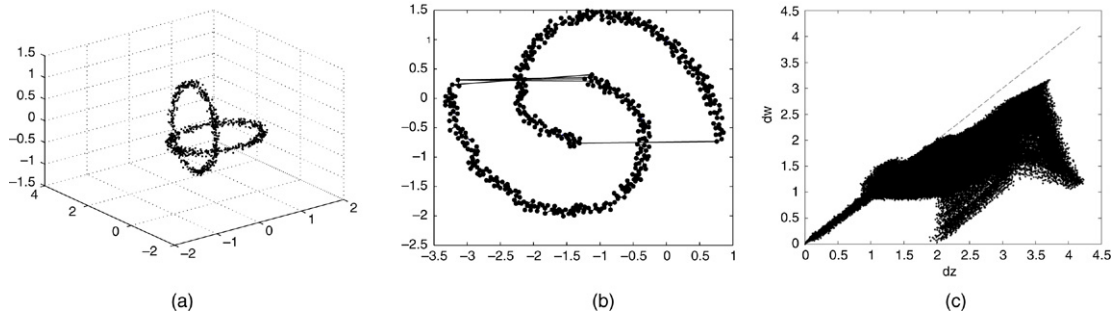

Fig. 1. (a) Chainlink dataset; (b) Chainlink quantized data projection with OVI-NG online and CHR connections; (c) "dz–dw" representation.

SOM-CCA offline. The term offline means that a two-step procedure is applied, where first SOM is used as a VQ method to learn the codebook vectors, and secondly CCA is used to project these codebook vectors in output space. A continuous output space is added to move the codebook positions freely, which is different from the conventional SOM's output layer. A variant is to change the quantizer to NG, to obtain the NG-CCA offline strategy.

In the SOM-CCA online version, SOM is used for training the codebook vectors in the input space, and simultaneously the projection part of CCA is applied for updating the codebook positions in the output space. Another variant is the NG-CCA online, which is obtained by changing the quantizer to NG.

The projection part of CCA is based on minimizing the following quadratic cost function:

$$E = \frac{1}{2}\sum_{j=1}^{N}\sum_{k\neq j}(D_{j,k} - d_{j,k})^2 F(D_{j,k}, \lambda_D), \tag{21}$$

where F is a weighting function, e.g. a step function, and λ_D is a user-controlled parameter that usually evolves in time. To minimize (21), a randomly chosen codebook position, z_k, is frozen, while all the others, $z_{j\neq k}$, are moved around it using the following update rule (Demartines & Hérault, 1997):

$$z_j(t+1) = z_j(t) + \alpha(t) F(D_{j,k}, \lambda_D) \times \frac{(D_{j,k} - d_{j,k})}{D_{j,k}}(z_k(t) - z_j(t)). \tag{22}$$

In addition, an offline version of our OVI-NG method was also considered. In the offline version, the codebook positions were updated after training the codebook vectors, i.e. the distances in input space were kept fixed.

3. Simulation results

In this section, the simulation results obtained with the OVI-NG online method are presented, and compared with the results obtained with the SOM-CCA and NG-CCA offline strategies, as well as the SOM-CCA and NG-CCA online strategies. An offline variant of our model, called OVI-NG offline, is also considered. In all simulations, the parameters of the OVI-NG algorithm were set as follows: $\epsilon_0 = 0.3$, $\epsilon_f = 0.001$, $\lambda_0 = 30.0$, $\lambda_f = 0.01$, $T_0 = 20.0$, and $T_f = 200.0$.

Four different data sets were considered: Chainlink, Iris, Sleep, and Fraud. The number of codebooks was set to 500, 70, 200 and 300, for each data set, respectively. The number of epochs of training for both NG and SOM was set as: Chainlink 500, Iris 1000, Sleep 500, and Fraud 500. The parameters α_0 and α_f were set as follows: Chainlink 0.3 and 0.1, Iris 0.9 and 0.3, Sleep 0.3 and 0.001, and Fraud 0.3 and 0.1, respectively. The parameters σ_0 and σ_f were set to the values: Chainlink 300 and 3, Iris 20 and 2, Sleep 150 and 0.1, Fraud 250 and 2, respectively.

The Chainlink data set consists of two torus-shaped clusters of 500 points each, which are intertwined like the links of a chain (Ultsch, 2005). The Iris data set contains 150 samples, with four attributes, distributed in three types of Iris flowers. The Sleep data set contains features extracted from polysomnographic recordings (Estévez et al., 2002). The Sleep data set contains 6463 samples, with six attributes and five classes. The Fraud data set has information about telephone subscribers (Estévez, Held, & Perez, 2006), and contains 10 624 samples, with 26 attributes and three classes.

Fig. 1 shows (a) the Chainlink data set, (b) the OVI-NG online output space, and (c) the "dz–dw" representation. It can be observed in Fig. 1(b) that OVI-NG has to break the links of the chain to be able to project them on a plane. This discontinuity is clearly visualized by the long connections established by CHR. Fig. 1(c) shows that there is a perfect distance preservation for distances lower than one.

Fig. 2 shows the (a) trustworthiness and (b) continuity measures as a function of the number of nearest neighbors, k, for the Chainlink data set. In Fig. 2(b), it can be observed that the best continuity is obtained by the OVI-NG method for all the k values explored. NG-CCA outperforms the original SOM-CCA in the whole range. The OVI-NG method also gets the best trustworthiness for $k < 20$, but it degrades for larger values of k, showing the tradeoff between trustworthiness and continuity. NG-CCA shows a high trustworthiness over the whole range of k values. Fig. 3 shows the topology preservation measure q_m as a function of the number of neighbors. High values of q_m indicate that the rank order is preserved within a local neighborhood of size k. Here the OVI-NG method shows a strong superiority over all the other methods for local neighborhoods containing fewer than 20 neighbors, but it is surpassed by the NG-CCA method for $k > 30$ neighbors. The worst q_m values were obtained with the original SOM-CCA method.

Fig. 4 shows the topographic function versus k for (a) online visualization methods and (b) offline visualization methods, for

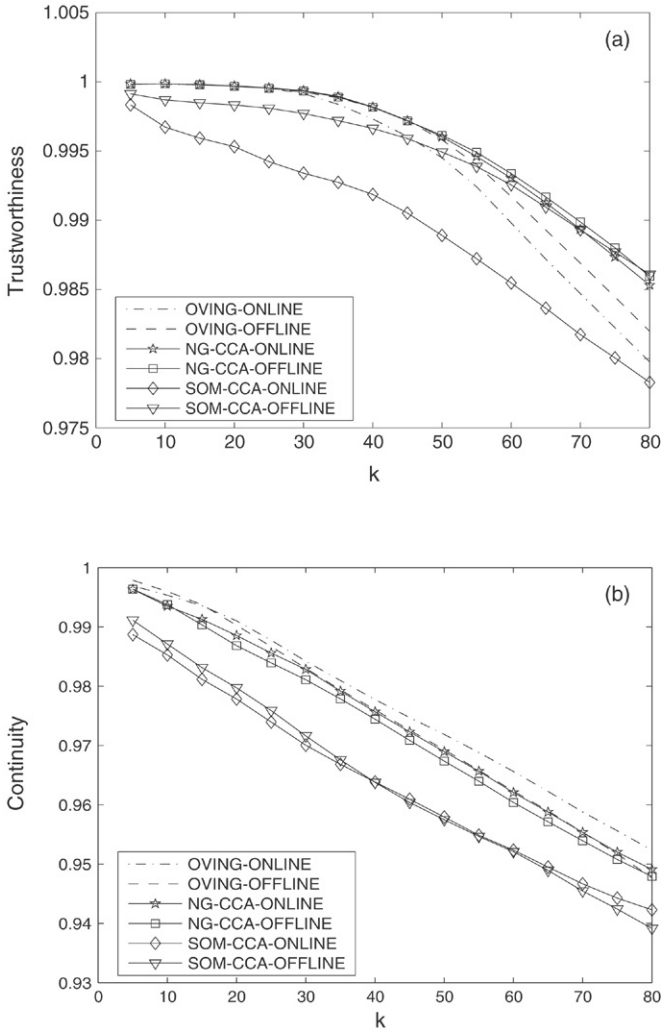

Fig. 2. (a) Trustworthiness and (b) continuity measures as a function of the number of neighbors, k, for the Chainlink data set.

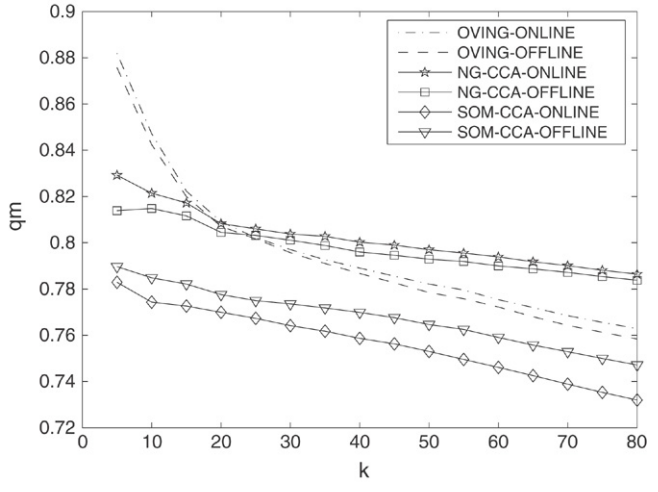

Fig. 3. Topology preservation measure q_m as a function of the number of neighbors, k, for the Chainlink data set.

the Chainlink data set. Both figures show that OVI-NG and NG-CCA have a similar behavior, outperforming SOM-CCA.

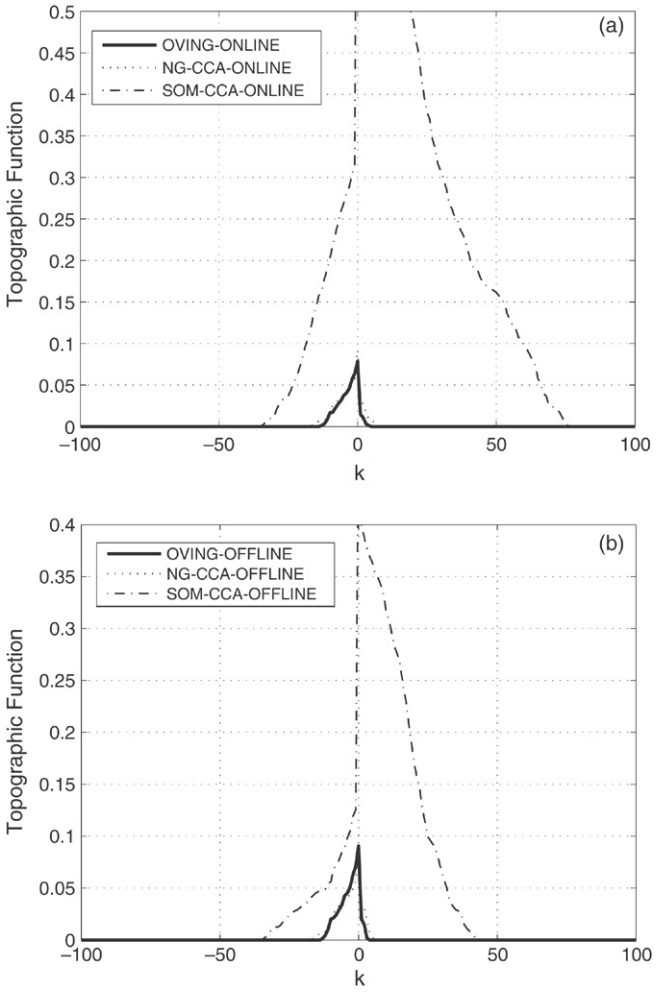

Fig. 4. Topographic function versus k for (a) online visualization methods, and (b) offline visualization methods, for the Chainlink data set.

Table 1
Topographic function value $\Phi(0)$ obtained with the different algorithms for the Chainlink data set, averaged over five simulations

Algorithm	$\Phi(0)$
OVING-ONLINE	0.0787 ± 0.0016
OVING-OFFLINE	0.0907 ± 0.0020
NG-CCA-ONLINE	0.0920 ± 0.0041
NG-CCA-OFFLINE	0.0921 ± 0.0028
SOM-CCA-ONLINE	1.0753 ± 0.3786
SOM-CCA-OFFLINE	0.5253 ± 0.3001

Table 1 shows that OVI-NG obtained the best topographic function value $\Phi(0)$ for the Chainlink data set.

Fig. 5 shows (a) the Iris quantized data projected with OVI-NG online, (b) the Iris data projection with CHR connections, and (c) the "dz–dw" representation. The position vectors were labeled with the majority class of the input vectors present in the Voronoi cell of the corresponding codebook vector. If the percentage of input vectors of the majority class is greater than a given threshold, the node is marked with the number of that class. Otherwise, the node is represented by a black ball with an exclamation sign inside. If the node wins no input vectors, it

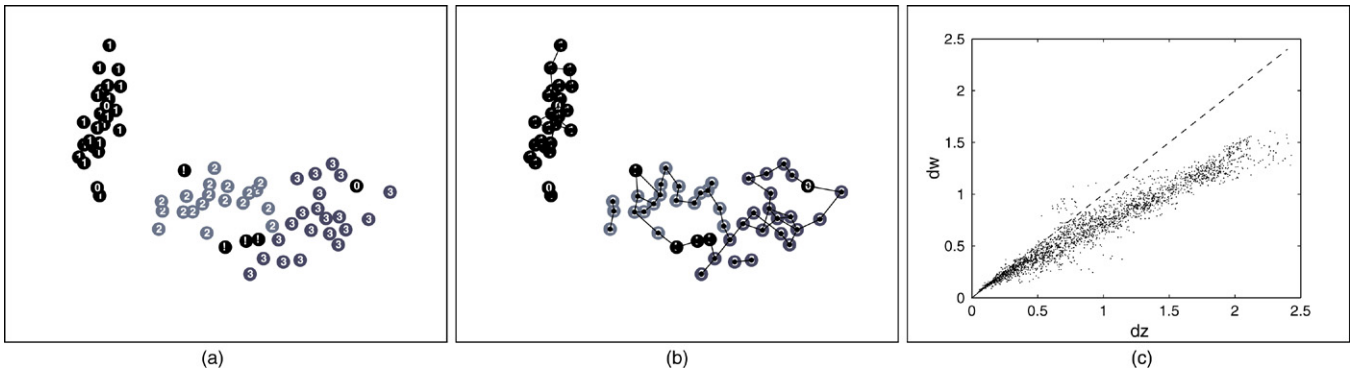

Fig. 5. (a) Iris quantized data projection with OVI-NG online, (b) Iris data projection with CHR connections, (c) "dz–dw" representation. The integer numbers represent the class of the majority of input vectors present in the Voronoi cell of the corresponding codebook vector, and 0 represents nodes that do not win any input vectors. The exclamation sign represent nodes associated to two or more classes.

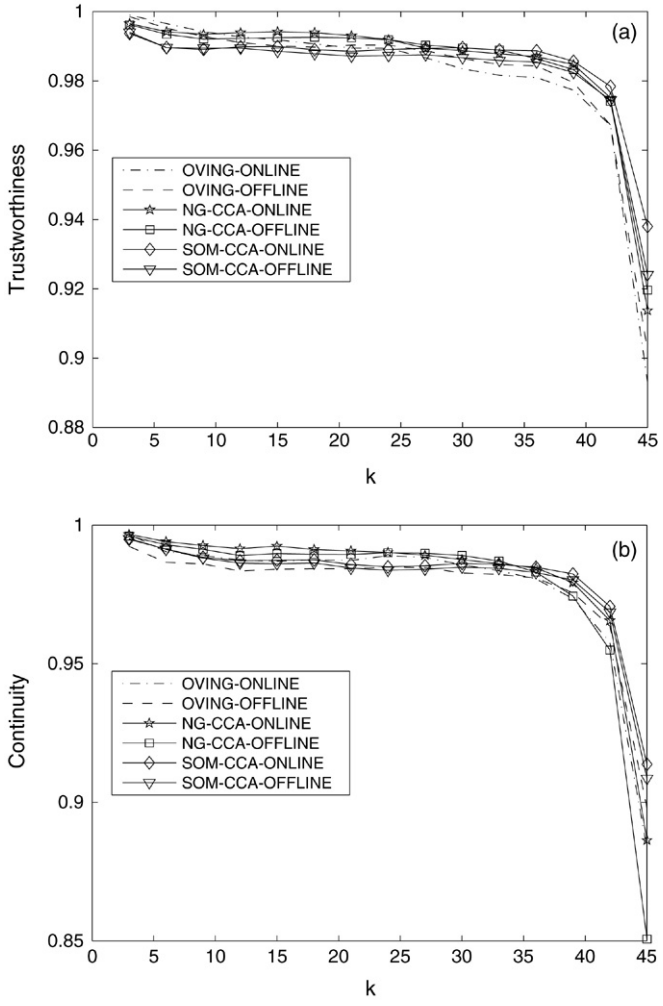

Fig. 6. (a) Trustworthiness and (b) continuity measures as a function of the number of neighbors, k, for the Iris data set.

is marked with a number 0. Fig. 5(c) shows that there is a very good distance preservation for distances smaller than 0.5.

Fig. 6 shows the (a) trustworthiness and (b) continuity as a function of the number of nearest neighbors, k, for the Iris data set. All methods show a high trustworthiness and continuity for k values lower than 30. In Fig. 6(a), it can be observed

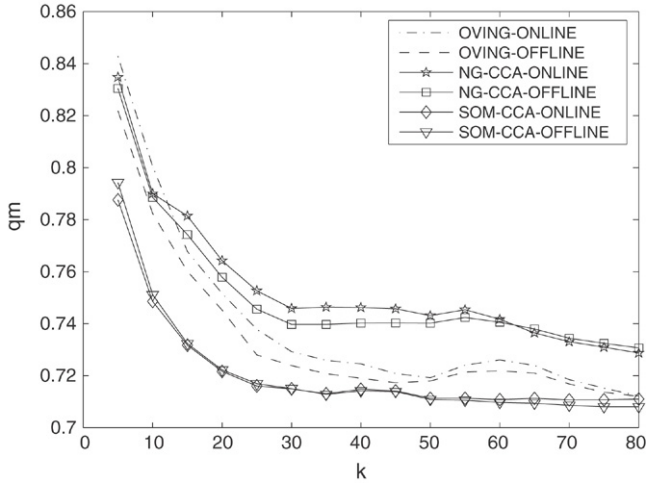

Fig. 7. Topology preservation measure q_m as a function of the number of neighbors, k, for the Iris data set.

that the best trustworthiness is obtained by OVI-NG online and NG-CCA for $k < 10$. Fig. 7 shows the topology preservation measure q_m as a function of the number of neighbors, for the Iris data set. Here, OVI-NG online outperforms all other methods for local neighborhoods containing fewer than 10 neighbors, but it is surpassed by the NG-CCA methods for higher values of k. The worst q_m values were obtained with the original CCA method.

Fig. 8 shows the topographic function versus k for (a) online visualization methods and (b) offline visualization methods, for the Iris data set. Both figures show that OVI-NG gets a lower peak at 0, and a faster decay to the right (positive values of k), outperforming other methods. Table 2 shows that OVI-NG obtained the best topographic function value $\Phi(0)$ for the Iris data set.

Fig. 9 shows (a) the Sleep quantized data projected with OVI-NG online, and (b) the Sleep data projection with CHR connections. Some discontinuities can be directly observed in Fig. 9(b) as links between pairs of points that are not nearest neighbors in the map.

Fig. 10 shows the (a) trustworthiness and (b) continuity measures as a function of the number of nearest neighbors,

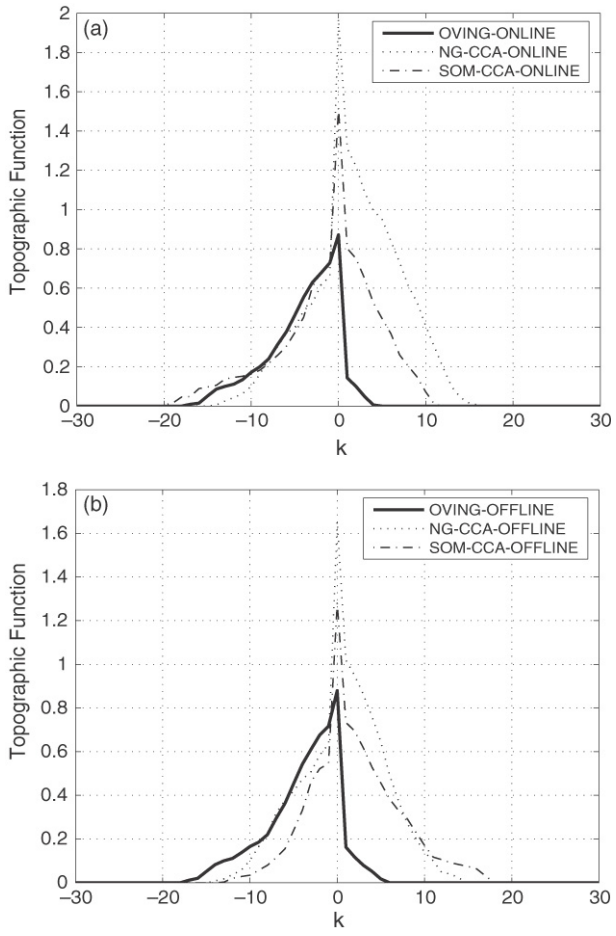

Fig. 8. Topographic function versus k for (a) online visualization methods, and (b) offline visualization methods, for the Iris data set.

k, for the Sleep data set. In Fig. 10(a), it can be observed that the best trustworthiness is obtained by NG-CCA offline, followed closely by the OVI-NG methods for $k < 60$. The latter predominates for $k > 60$. The OVI-NG and NG-CCA offline methods obtained the best continuity for k values lower than 40. Fig. 11 shows the topology preservation measure q_m as a function of the number of neighbors for the Sleep data set. Here, NG-CCA outperforms all other methods, OVI-NG being the second best. The worst q_m values were obtained with the original SOM-CCA method.

Table 2
Topographic function value $\Phi(0)$ obtained with the different algorithms for the Iris dataset, averaged over five simulations

Algorithm	$\Phi(0)$
OVING-ONLINE	0.8714 ± 0.0881
OVING-OFFLINE	0.8800 ± 0.0931
NG-CCA-ONLINE	1.9714 ± 0.4236
NG-CCA-OFFLINE	1.6543 ± 0.4615
SOM-CCA-ONLINE	1.5171 ± 0.8248
SOM-CCA-OFFLINE	1.2800 ± 0.9562

Table 3
Topographic function value $\Phi(0)$ obtained with the different algorithms for the Sleep data set, averaged over five simulations

Algorithm	$\Phi(0)$
OVING-ONLINE	1.0001 ± 0.5594
OVING-OFFLINE	0.9903 ± 0.6374
NG-CCA-ONLINE	3.1567 ± 2.1569
NG-CCA-OFFLINE	3.4762 ± 2.5901
SOM-CCA-ONLINE	2.9038 ± 2.6997
SOM-CCA-OFFLINE	1.5904 ± 0.4044

Fig. 12 shows the topographic function versus k for (a) online visualization methods and (b) offline visualization methods, for the Sleep data set. Both figures show that OVI-NG gets lower values of Φ in almost the entire range of k, outperforming both SOM-CCA and NG-CCA. Table 3 shows that OVI-NG obtained the best topographic function value $\Phi(0)$ for the Sleep data set.

Fig. 13 shows the Fraud quantized data projection obtained with OVI-NG online, and Fig. 14 shows the Fraud data projection with CHR connections. Some discontinuities can be observed in Fig. 14 as links between pairs of points that are not nearest neighbors in the map.

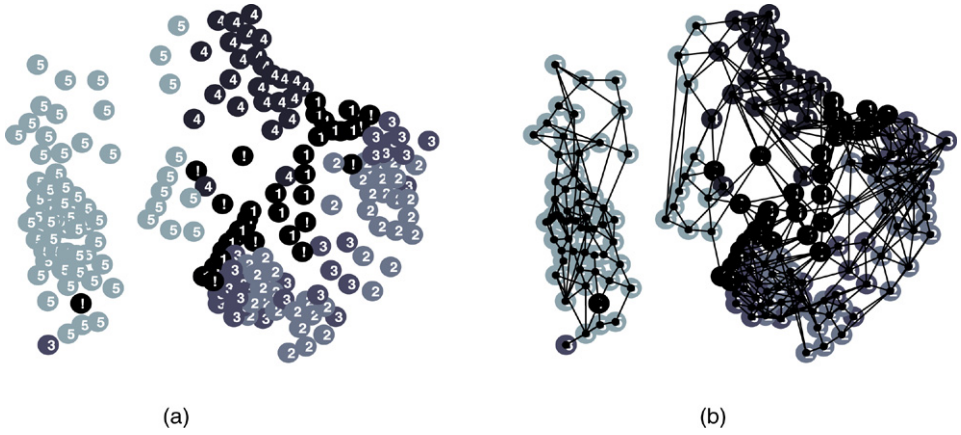

Fig. 9. (a) Sleep quantized data projection with OVI-NG online; (b) Sleep data projection with CHR connections. The integer numbers represent the class of the majority of input vectors present in the Voronoi cell of the corresponding codebook vector, and 0 represents nodes that do not win any input vectors. The exclamation sign represents nodes associated with two or more classes.

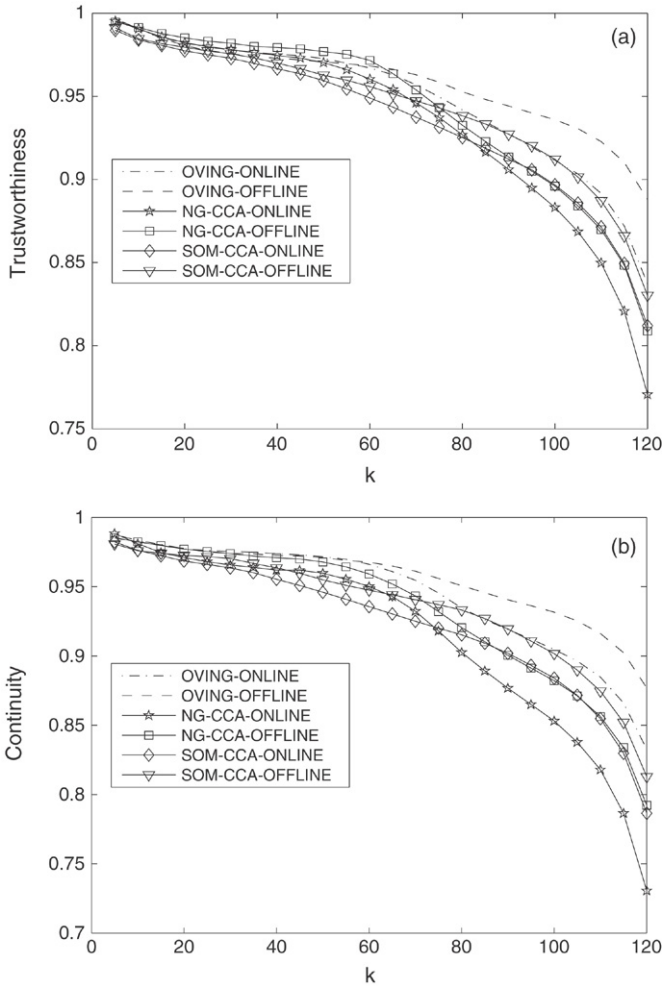

Fig. 10. (a) Trustworthiness and (b) continuity measures as a function of the number of neighbors, k, for the Sleep data set.

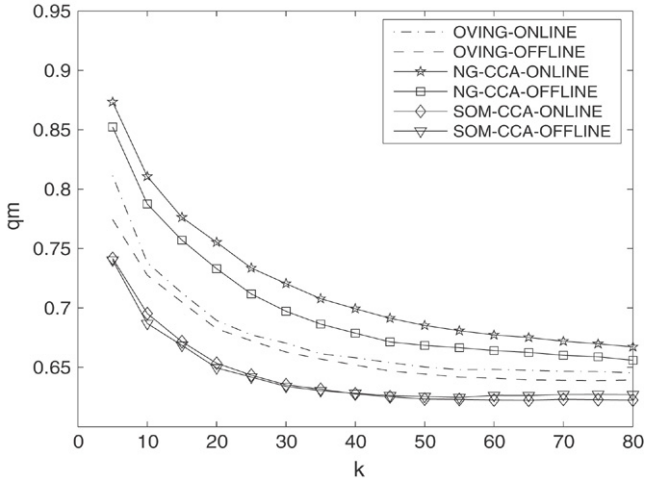

Fig. 11. Topology preservation measure q_m as a function of the number of neighbors, k, for the Sleep data set.

Fig. 15 shows the (a) trustworthiness and (b) continuity measures as a function of the number of nearest neighbors, k, for the Fraud data set. It can be observed that the best trustworthiness and continuity is obtained by the OVI-NG

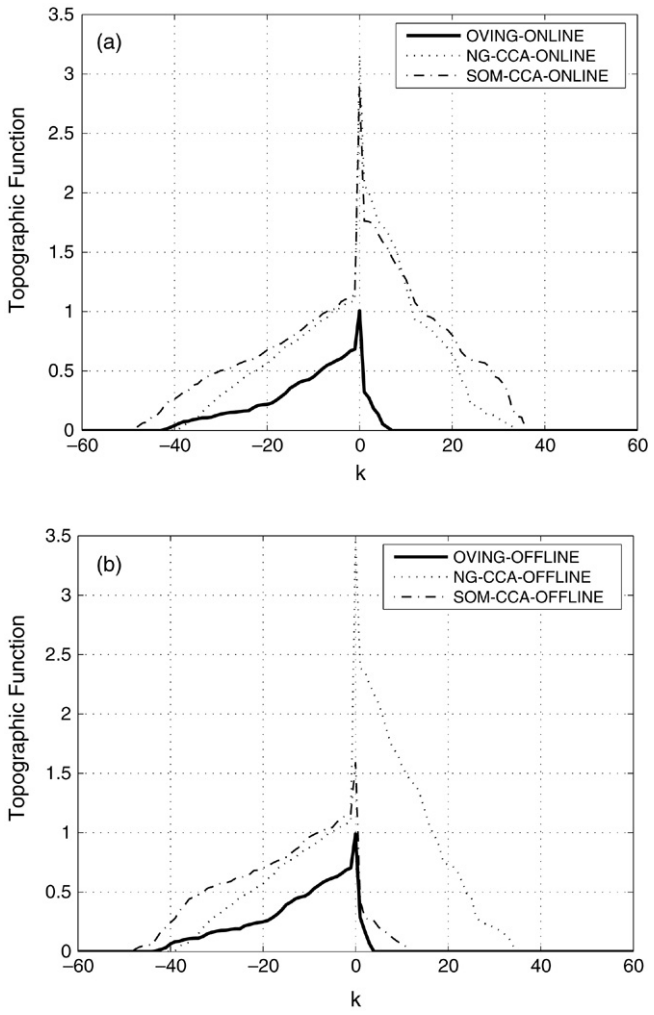

Fig. 12. Topographic function versus k for (a) online visualization methods, and (b) offline visualization methods, for the Sleep data set.

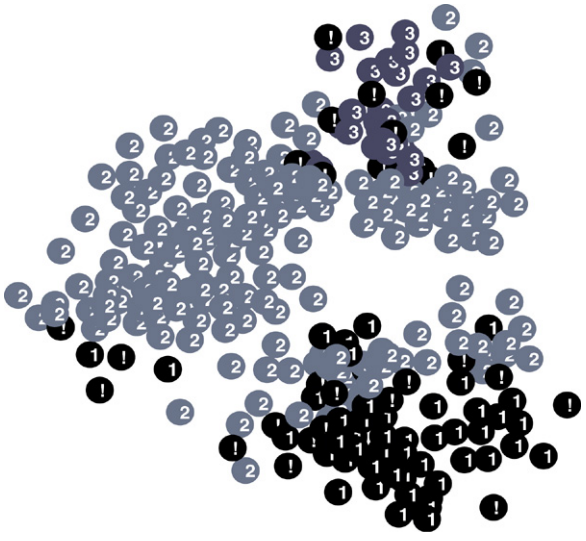

Fig. 13. Fraud quantized data projection with OVI-NG online. The integer numbers represent the class of the majority of input vectors present in the Voronoi cell of the corresponding codebook vector, and 0 represents nodes that do not win any input vectors. The exclamation sign represents nodes associated with two or more classes.

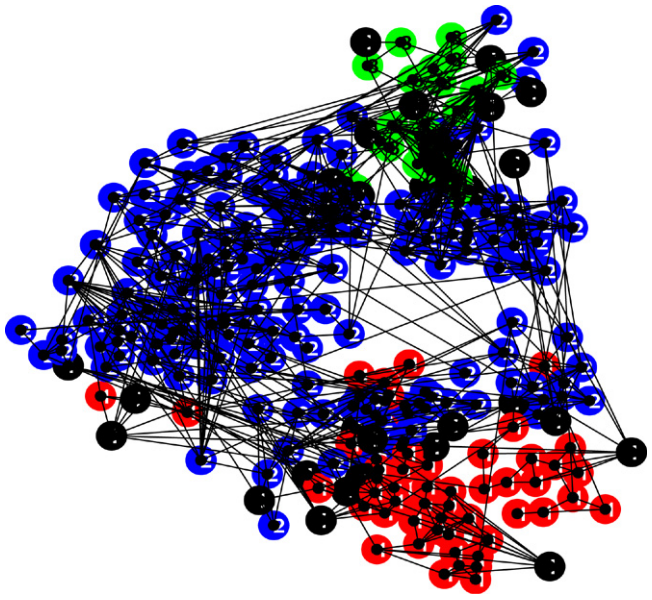

Fig. 14. Fraud data projection with CHR connections.

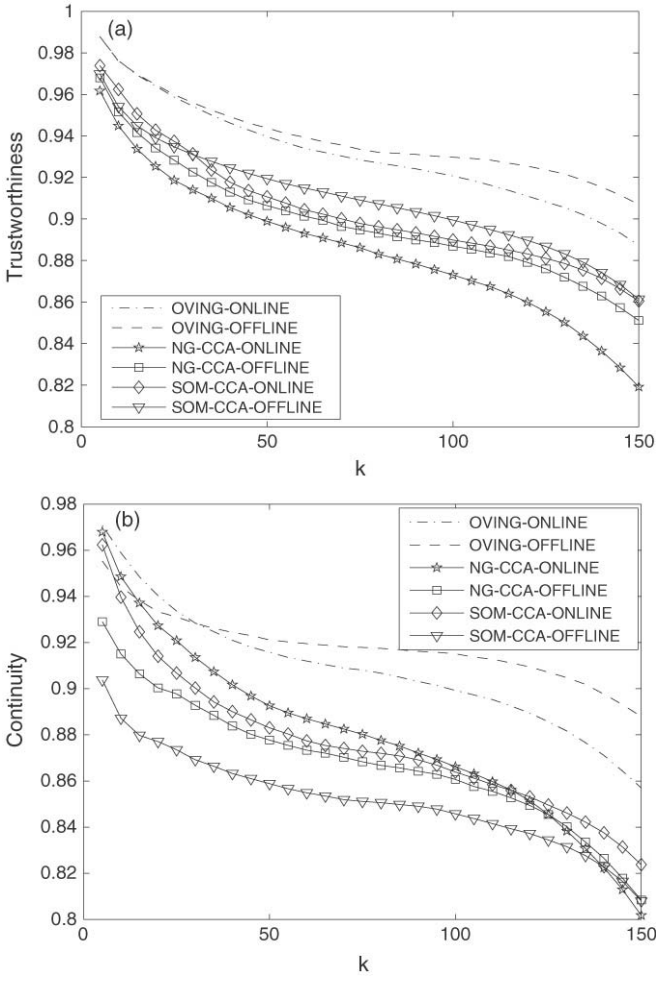

Fig. 15. (a) Trustworthiness and (b) continuity measures as a function of the number of neighbors, k, for the Fraud data set.

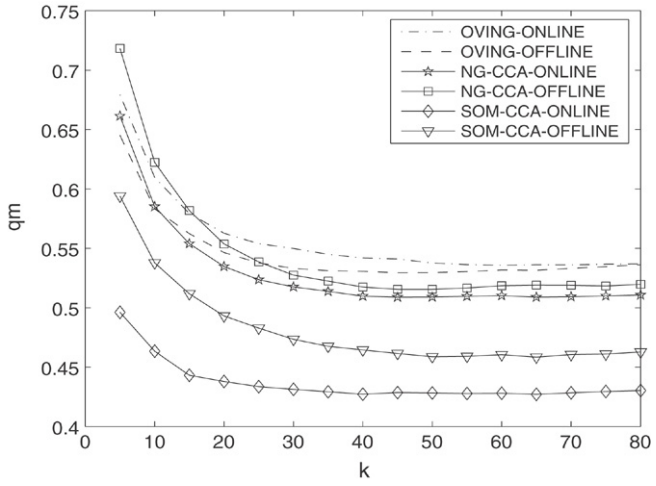

Fig. 16. Topology preservation measure q_m as a function of the number of neighbors, k, for the Fraud data set.

Table 4
Topographic function value $\Phi(0)$ obtained with the different algorithms for the Fraud data set, averaged over five simulations

Algorithm	$\Phi(0)$
OVING-ONLINE	2.5089 ± 0.2798
OVING-OFFLINE	2.3117 ± 0.0589
NG-CCA-ONLINE	7.1278 ± 1.7608
NG-CCA-OFFLINE	2.8564 ± 0.6788
SOM-CCA-ONLINE	7.6702 ± 1.8174
SOM-CCA-OFFLINE	2.6184 ± 0.8869

methods for all k values explored. Here, NG-CCA obtained the worst trustworthiness. Fig. 16 shows the topology preservation measure q_m as a function of the number of neighbors for the Fraud data set. Here the OVI-NG methods outperform all other methods, except for $k < 15$, where NG-CCA offline slightly predominates. The worst q_m values were obtained with the original SOM-CCA method.

Fig. 17 shows the topographic function versus k for (a) online visualization methods and (b) offline visualization methods, for the Fraud data set. Both figures show that OVI-NG obtains lower values of Φ for almost the whole range of k, outperforming both SOM-CCA and NG-CCA. Table 4 shows that OVI-NG obtained the best topographic function value $\Phi(0)$ for the Fraud data set.

4. Discussions and conclusions

The tradeoff between trustworthiness and continuity may be controlled by adding to Eq. (5) a term that penalizes errors in preserving distances in the input space, as has been proposed in Venna and Kaski (2005). In such a case, the cost function could be changed to

$$E = \frac{1}{2}\sum_{j=1}^{N}\sum_{k\neq j}(D_{j,k} - d_{j,k})^2 G(r_{j,k}, s_{j,k})$$

$$= \frac{1}{2}\sum_{j=1}^{N}\sum_{k\neq j} E_{j,k}, \qquad (23)$$

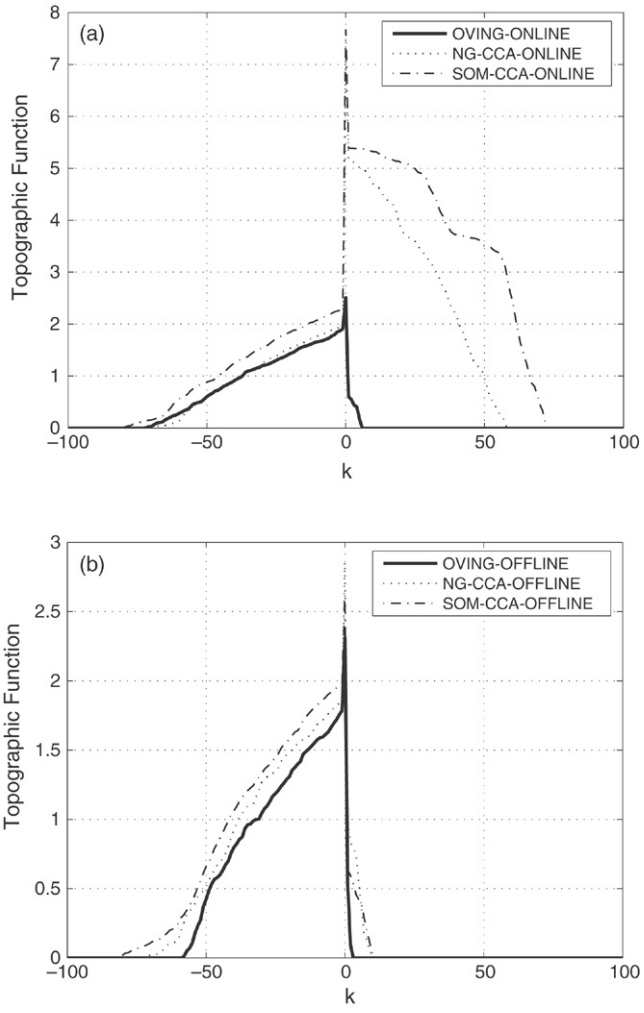

Fig. 17. Topographic function versus k for (a) online visualization methods, and (b) offline visualization methods, for the Fraud data set.

where

$$G(r_{j,k}, s_{j,k}) = \big((1-\beta) \times F(s_{j,k})\big) + \beta \times F(r_{j,k}). \qquad (24)$$

The parameter β allows us to control the tradeoff between trustworthiness and continuity of the map. In this version of the algorithm, the function F must be replaced by G in Eqs. (7) and (8). Fig. 18 shows how the trustworthiness and continuity measures change when varying the parameter β for the Iris data set. The maximum trustworthiness is obtained for $\beta = 0$, which corresponds to the OVI-NG algorithm. When increasing β from 0 to 0.7, the continuity measure increases but the trustworthiness decreases, showing clearly the trade-off between these two measures. The minimum trustworthiness is obtained for $\beta = 1$, which corresponds to the case when the function G depends only on the neighborhood ranking in input space.

If the projection were trustworthy, then the problem would be reduced to discontinuities in the map. The display of CHR connections may help to visualize these discontinuities. A better display could be devised by showing only connections among non-neighboring units.

The proposed OVI-NG method provides an output representation to the neural gas model that is useful for data

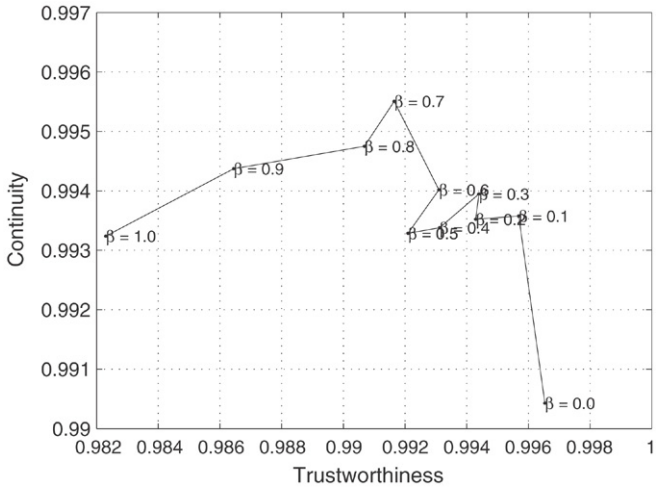

Fig. 18. Continuity versus trustworthiness when varying the value of the parameter β, for the Iris data set.

projection and visualization. The OVI-NG is an online method that concurrently adjusts the codebook vectors in input space and the codebook positions in a continuous output space. The method is computationally efficient with $O(N \log N)$ complexity. The position adaptation rule minimizes a cost function similar to CCA, which favors the local topology.

For several data sets, the OVI-NG method outperformed the original SOM-CCA method and its NG-CCA variant, in both their offline and online versions, in terms of trustworthiness, continuity, topographic function and local preservation measures, for neighborhood sizes smaller than 20 or 30. The results show empirically that NG is a better quantizer than SOM, since, in general, NG-CCA obtained much better performances than SOM-CCA for all data sets.

In future research, OVI-NG could be extended to map all data points in an efficient way, using the codebook vectors and their projection positions as a reference. Another possible extension is to incorporate curvilinear or graph distances, in order to unfold strongly nonlinear maps.

Acknowledgment

This research was supported by Conicyt-Chile, under grant Fondecyt 1050751.

References

Cox, T. F., & Cox, M. A. A. (2001). *Multidimensional scaling* (2nd ed.). Boca Raton: Chapman and Hall/CRC.

Demartines, P., & Hérault, J. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, *8*(1), 148–154.

Estévez, P. A., & Figueroa, C. J. (2005). Online nonlinear mapping using the neural gas network. In *Proceedings of the workshop on self-organizing maps* (pp. 299–306).

Estévez, P. A., Figueroa, C. J., & Saito, K. (2005a). Cross-entropy embedding of high dimensional data using the neural gas model. *Neural Networks*, *18*(5–6), 727–737.

Estévez, P. A., Figueroa, C. J., & Saito, K. (2005b). Cross-entropy approach to data visualization based on the neural gas network. In *Proceedings of the international joint conference on neural networks* (pp. 2724–2729).

Estévez, P. A., Held, C. M., Holzmann, C. A., Perez, C. A., Pérez, J. P., Heiss, J., Garrido, M., & Peirano, P. (2002). Polysomnographic pattern recognition for automated classification of sleep-waking states in infants. *Medical and Biomedical Engineering and Computing*, *40*, 105–113.

Estévez, P. A., Held, C. M., & Perez, C. A. (2006). Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. *Expert Systems with Applications*, *31*, 337–344.

Figueroa, C. J., Estévez, P. A., & Hernández, R. (2005). Non-linear mappings based on particle swarm optimization. In *Proceedings of the international joint conference on neural networks* (pp. 1487–1492).

Figueroa, C. J., & Estévez, P. A. (2004). A new visualization scheme for self-organizing neural networks. In *Proceedings of the international joint conference on neural networks* (pp. 757–762).

Kaski, S., Nikkilä, J., Oja, M., Venna, J., Törönen, P., & Castrén, E. (2003). Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, *4*, 48.

Kohonen, T. (1995). *Self-organizing maps*. Berlin, Germany: Springer-Verlag.

König, A., & Michel, T. (2003). DIPOL-SOM—A distance preserving enhancement of the self-organizing map for dimensionality reduction and multivariate data visualization. In *Proceedings of the workshop on self-organizing maps* (pp. 219–224).

König, A. (2000). Interactive visualization and analysis of hierarchical neural projections for data mining. *IEEE Transactions on Neural Networks*, *11*(3), 615–624.

Lee, J. A., Lendasse, A., & Verleysen, M. (2004). Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, *57*, 49–76.

Lee, J. A., Lendasse, A., Donckers, N., & Verleysen, M. (2000). A robust nonlinear projection method. In *Proceedings of the European symposium on artificial neural networks* (pp. 13–20).

Marsland, S., Shapiro, J., & Nehmzow, U. (2002). A self-organizing network that grows when required. *Neural Networks*, *15*, 1041–1058.

Martinetz, T. M., & Schulten, K. J. (1994). Topology representing networks. *Neural Networks*, *7*(3), 507–522.

Martinetz, T. M., Berkovich, S. G., & Schulten, K. J. (1993). Neural gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, *4*, 558–569.

Martinetz, T. M., & Schulten, K. J. (1991). A neural-gas network learns topologies. In T. Kohonen, K. Makisara, O. Simula, & J. Kangas (Eds.), *Artificial neural networks* (pp. 397–402). Amsterdam: North-Holland.

Merkl, D., & Rauber, A. (1997). Alternative ways for cluster visualization in self-organizing maps. In *Proceedings of the workshop on self-organizing maps* (pp. 106–111).

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by local linear embedding. *Science*, *290*, 2323–2326.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, *C18*, 401–409.

Su, M. C., & Chang, H. T. (2001). A new model of self-organizing neural networks and its application in data projection. *IEEE Transactions on Neural Networks*, *12*, 153–158.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Ultsch, A. (2005). Clustering with SOM: U*C. In *Proceedings of the workshop on self-organizing maps* (pp. 75–82).

Ultsch, A. (2003). Maps for the visualization of high-dimensional data spaces. In *Proceedings of the workshop on self-organizing maps* (pp. 225–230).

Ultsch, A., & Siemon, H. P. (1990). Kohonen's self-organizing feature maps for exploratory data analysis. In *Proceedings of the international neural network conference* (pp. 305–308).

Venna, J., & Kaski, S. (2005). Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In *Proceedings of the workshop on self-organizing maps* (pp. 695–702).

Villmann, T., Der, R., Herrmann, M., & Martinetz, T. M. (1997). Topology preservation in self-organizing feature maps: exact definition and measurement. *IEEE Transactions on Neural Networks*, *8*(2), 256–266.

Yin, H. (2003). Resolution enhancement for the ViSOM. In *Proceedings of the workshop on self-organizing maps* (pp. 208–212).

Yin, H. (2002a). Data visualization and manifold mapping using the VISOM. *Neural Networks*, *15*, 1005–1016.

Yin, H. (2002b). ViSOM — A novel method for multivariate data projection and structure visualization. *IEEE Transactions on Neural Networks*, *13*, 237–243.