# Nonlinear principal component analysis

[1]Ralf Der, [1]Ulrich Steinmetz, [1]Gerd Balzuweit, and [2]Gerrit Schüürmann

[1] University of Leipzig, Institute of Informatics *

[2] Umweltforschungszentrum Leipzig-Halle GmbH

June 26, 1998

**Abstract**

We study the extraction of nonlinear data models in high-dimensional spaces with modified self-organizing maps. We present a general algorithm which maps low-dimensional lattices into high-dimensional data manifolds without violation of topology. The approach is based on a new principle exploiting the specific dynamical properties of the first order phase transition induced by the noise of the data. Moreover we present a second algorithm for the extraction of generalized principal curves comprising disconnected and branching manifolds. The performance of the algorithm is demonstrated for both one- and two-dimensional principal manifolds and also for the case of sparse data sets. As an application we reveal cluster structures in a set of real world data from the domain of ecotoxicology.

## 1 Introduction

One of the major objectives of data analysis is the extraction and instructive representation of the relevant information contained in the data. In cases of practical interest, data are given by high-dimensional vectors corrupted by noise. Dimension reduction and elimination of noise then is the essential step in analyzing the data. Principal component analysis (PCA) is one of the most prominent tools in this process. By uncovering the principal components of the data distribution PCA creates a lower dimensional subspace which contains the relevant information on the data.

Although highly successful in typical cases PCA suffers from the drawback of being a linear method. By way of example consider the globe with the locations of cities as data points. PCA would discover in this data set three principal components so that there is no complexity reduction in the description of the data. On the other hand a topographic map of the globe provides a two dimensional representation which can be analysed successfully using conventional methods like PCA.

*P. O. Box 920, 04109 Leipzig, Germany. e-mail: der@informatik.uni-leipzig.de or ulrich@mis.mpg.de

The location of cities on the globe forms a nonlinear data manifold. The above example suggests a two-step treatment by first mapping the nonlinear data set onto a linear lower-dimensional manifold and using conventional methods after. However real-world data manifolds besides of being nonlinear often are corrupted by noise and embed into high dimensional spaces. The present paper will present general procedures for finding the optimal mappings in this general case.

Looking into the opposite direction, the map can also be seen as embedding a low-dimensional manifold $\mathcal{M}$, a regular lattice, say, into the higher dimensional data manifold. $\mathcal{M}$ is called a principal manifold (PM) of the data if it provides an optimized (in some sense) representation of the data. A convenient choice is defined self-consistently by the requirement that each point on the PM is the average of the data points projecting to it, cf. [8]. Thus, it minimizes the mean square deviations of the data from the PM subject to some smoothness constraint.

The present contribution provides general procedures for finding such principal manifolds for arbitrary data sets.

# 2 Principal curves

Let us now consider the problem of finding principal manifolds in some detail. We will restrict to the case of principal curves, since this does already show the full complexity of the problem.

## 2.1 Definition of principal curves

Let us consider a data set $\mathcal{X}$ with data $\mathbf{v} = (v_1, \ldots, v_n) \in \mathcal{R}^n$. A principal component $\mathcal{P}$ describes a data set $\mathcal{X}$ as linear function $\mathbf{f}$ of a single parameter $\lambda$ i.e. $\mathbf{v} \in \mathcal{X}$ is represented by $\mathbf{f}(\mathbf{v}) = \lambda(\mathbf{v})\ \mathbf{c} + \mathbf{c}_0 \quad \in \mathcal{P}$. Given $\mathcal{X}$, the vectors $\mathbf{c}$ and $\mathbf{c}_0$ are determined by minimizing the reconstruction error

$$\mathbf{c_0}, \mathbf{c}: \qquad \frac{\partial}{\partial \mathbf{c_0}} E = 0, \quad \frac{\partial}{\partial \mathbf{c}} E = 0 \tag{1}$$

where

$$E = \int_{\mathcal{X}} \|\mathbf{v} - \lambda \mathbf{c} - \mathbf{c}_0\|^2 P(\mathbf{v}) d\mathbf{v}. \tag{2}$$

$P(\mathbf{v})$ being the probability distribution of the data. Eqs. (1),(2) imply that the distance $\|\mathbf{v} - \mathbf{w}\|$ is minimal with respect to variations of $\mathbf{w}$ along $\mathcal{P}$. In other words, the projection of a data point $\mathbf{v}$ is given by its closest point on the principal component.

A principal curve $\mathcal{P}_{\mathcal{X}}$ is a generalized principal component in that a large class of **nonlinear** smooth vector-valued function $\mathbf{f}(\lambda)$, $\lambda \in R^1$ is allowed for the representation of the data. The projection $\lambda(\mathbf{v})$ of a data point $\mathbf{v}$ onto the curve is again defined by the value of $\lambda$ for which $\mathbf{f}(\lambda)$ is closest to $\mathbf{v}$

$$\lambda(\mathbf{v}) = \arg\min_{\lambda} (\mathbf{v} - \mathbf{f}(\lambda))^2 \tag{3}$$

In general each point (segment in the discrete case) of the curve is the projection of a subset of data points called its projectors. In the linear case the projections agree with the center of gravity of the projectors. In this sense one may say that the principal component is running through the middle of its data points. This definition can be carried over to the nonlinear case. Hence a principal curve is defined as running through the center of gravities of the points projecting to it, see Fig. 1. One may also say that each point on the PC is required to be the average of its projections. This can be

r

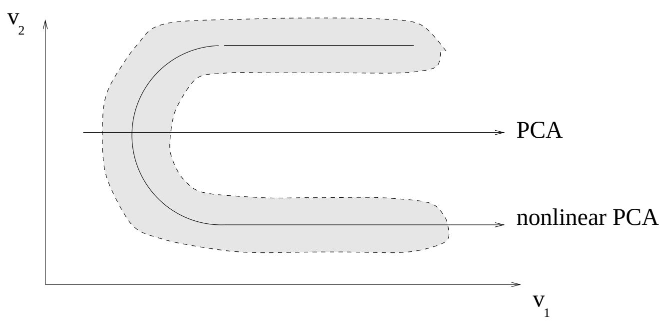


Figure 1: Principal component and principal curve. The gray region depicts a nonlinear data manifold in data space $\mathcal{X}$. The principal component (PCA) ignores the structure of the data distribution. The principal curve (nonlinear principal component) provides a faithful one-dimensional representation of the two-dimensional nonlinear data manifold.

formulated in terms of a variational principle, i. e. we define the mean square deviation between data points and its projections (reconstruction error)

$$E = \int d\mathbf{v} P(\mathbf{v})(\mathbf{v} - \mathbf{f}(\lambda))^2 \tag{4}$$

and require its variation with respect to $\mathbf{f}(\lambda)$ to be zero.

$$\frac{\delta}{\delta \mathbf{f}} E = 0 \tag{5}$$

Note that we must not require the reconstruction error to be minimal as in the linear case. Instead the weaker property of stationarity arises as implied by eq. (5). Why this happens is best seen from the pathological case of a curve that runs through every point of a noisy data set $\mathcal{X}$ so that it has a zero reconstruction error. However this curve does not fit the purpose of a principal curve since it provides no elimination of noise. In order to prevent such pathologies an additional condition is required for completing the definition of a principal curve. Usually this is a smoothness condition which restricts the curvature of the PC. Another condition is the requirement that the map $\mathbf{v} \to \lambda$ should be topographic in the best possible way. It is the precise formulation of these criteria which makes the problem highly nontrivial.

## 2.2 The Hastie-Stuetzle algorithm

Hastie and Stuezle [8] described an algorithm for the construction of principal curves which works iteratively starting from the principal component of the data set. In each iteration a new estimate of the PC is obtained from the calculation of the centers of gravity of the data points with respect to the current estimate of the PC. This procedure is combined with a smoothing operation controlled via cross validation in order to avoid the over-fitting catastrophe. The authors gave some evidence in favor of the convergence to a stable solution, mainly by referring to the linear case. However, so far there are no general criteria for the existence and uniqueness of the PC. Hastie and Stuetzle succeeded in showing that under the smoothness constraint, with respect to the reconstruction error, the PM is a stationary point in function space. However no general results exists as to the stability of this solution.

In the Hastie-Stuetzle algorithm and other algorithms known so far the smoothness and hence the stability are guaranteed by local averaging, the span of the average being guided **globally** by cross validation. We will provide a new algorithm below which avoids this restriction and moreover leads to a stable though possibly suboptimal principal curve. Moreover the new algorithm is not restricted to the case of principal **curves**.

# 3 Self-organizing maps and principal manifolds

Topographic maps are a fundamental functional unit of neural information processing systems. These maps are learned during individual development from the data stream. There are several algorithms modeling this unsupervised learning process. Let us consider Kohonen's algorithm first which self–organizes topographic mappings between manifolds embedded in spaces of different dimensionality guaranteeing a good control of the smoothness of the maps. The application to the problem of finding principal manifolds has been discussed before, cf. [15]. We present this approach first and introduce subsequently a new method which is distinct by its self-regulating, local control of smoothness.

## 3.1 Kohonen's algorithm

Let us assume that the data embedded in $d$-dimensional space $\mathcal{X}$ may be viewed as a data cloud scattering about some manifold of the lower dimension $D < d$,. Then Kohonen's algorithm may be used to map the data topographically onto a $D$-dimensional lattice $\mathcal{A}$, where the lattice sites $\mathbf{r} \in [1, N]^D$ may be considered as the physical positions of $N^D$ neurons, see Fig. 2 for a generic example. Upon presentation of a data vector $\mathbf{v}$, the shift $\Delta \mathbf{w_r}$ for the synaptic vectors $\mathbf{w_r}$ is

$$\Delta \mathbf{w_r} = \varepsilon \, h_{\mathbf{r},\mathbf{r}'} \left(\mathbf{v} - \mathbf{w_r}\right) , \tag{6}$$

where the position $\mathbf{r}'$ of the winner (best match or closest) neuron is determined by

$$\mathbf{r}' = \arg\max_{\mathbf{r}} \|\mathbf{v} - \mathbf{w_r}\| \, . \tag{7}$$

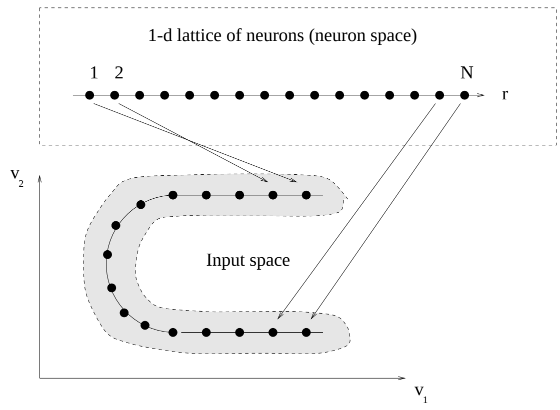


Figure 2: Finding the nonlinear principal component by a self–organizing neural map. The neuron space (consisting of a chain of neurons in the present case) is depicted in the dashed box above. Neuron positions are at lattice sites $r_n$ where $n = 1, 2, \ldots, N$. The $2-d$ input space $\mathcal{X}$ (bottom) contains the distribution (gray region) of the data vectors $\mathbf{v} = (v_1, v_2)$. Each neuron $n$ carries a synaptic vector $\mathbf{w}_n = (w_{n1}, w_{n2})$ which is represented also in input space by a black dot called the projection or virtual position of the neuron in input space. These (virtual) positions of the neurons are the supports for the principal curve (solid line) running through the middle of the data distribution . Data points are mapped to (virtual positions of) neurons by the principle of shortest distance. The map obtained in the present case is a neighborhood preserving or topographic one.

which means nothing else than that the distance between $\mathbf{v}$ and the virtual position $\mathbf{w}_{\mathbf{r}'}$ of neuron at lattice site $\mathbf{r}'$ is not larger than the distance of $\mathbf{v}$ to any of the other neurons. The neighborhood function

$$h_{\mathbf{r},\mathbf{r}'} = \left(2\pi\sigma^2\right)^{-D/2} \exp\left(-(\mathbf{r}-\mathbf{r}')^2/(2\sigma^2)\right) \tag{8}$$

defines the range of cooperativity between neurons. The map is initialized by choosing random values for the synaptic vectors, see Fig. 3.

The neighborhood function, cf. eq. 8 controls the smoothness of the map. The argument is that the curve in input space of maximum curvature that can be formed by a set of neurons which are situated in a region of length $2\sigma$ on the chain is essentially a semi-circle. Hence the local radius of curvature is obtained by finding two neurons which are a distance $\|\mathbf{r_1} - \mathbf{r_2}\| \approx \sigma$ apart in the neuron space $\mathcal{A}$. Then the local radius $\rho$ of curvature is not less than $\rho_{min} = 2\|\mathbf{w}_{\mathbf{r_1}} - \mathbf{w}_{\mathbf{r_2}}\|$.

However, in practical applications one does not know the appropriate smoothness for an optimal principal manifold. Moreover the smoothness is controlled only globally in Kohonen's algorithm. This is a serious drawback with locally varying widths of the scattering of noisy data points around the principal manifolds. The latter situation would require the smoothness of the map to be controlled locally as a function of the scattering of the data.

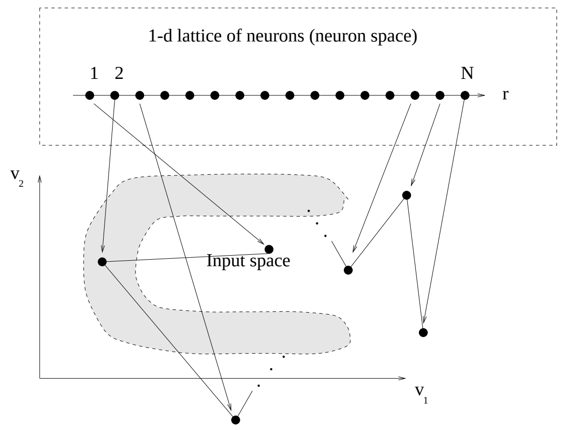


Figure 3: The initial map obtained by choosing random values for the components of the synaptic vectors. The picture shows both the virtual positions and the corresponding lattice bonds for the first and last three neurons on the chain.

## 3.2 Local self–control of smoothness

Local control of smoothness may be achieved by an individual neighborhood $\sigma_{\mathbf{r}}$ for each neuron, so that

$$h_{\mathbf{r},\mathbf{r}'} = \left(\frac{1}{\sqrt{2\pi}\tilde{\sigma}_{\mathbf{r}}}\right)^{D} \exp\left(-\frac{(\mathbf{r}-\mathbf{r}')^2}{2\tilde{\sigma}_{\mathbf{r}}^2}\right) \tag{9}$$

where $\tilde{\sigma}_{\mathbf{r}} = \min(\sigma_{\mathbf{r}}, \sigma_{\mathbf{r}'})$ with the additional constraints $1 \leq \tilde{\sigma}_{\mathbf{r}} \leq \sigma_{\max}$. [1]

The crucial point now is the determination of the local values $\sigma_{\mathbf{r}}$. For this purpose we exploit the dynamics of the phase transition from the topographic map to the over-fitting situation, cf. [15, 6]. For a discussion consider the case of mapping a chain of neurons into a data manifold of dimension higher than one as demonstrated in Fig. 2. The algorithm tries to adapt the (image of) the chain to the data points as close as possible under the smoothness constraint defined by the value of $\sigma$. While gradually shrinking $\sigma$ the chain adapts closer and closer to the data points ending up with a complete match to all the data points which is the over-fitting catastrophe.

The point now is that there is a sharp phase transition to the over-fitting regime which occurs at a critical value $\sigma_c$ of the neighborhood width, $\sigma_c$ depending on the scattering of the data points about the principal curve. At the phase transition point the quality of the map changes in that characteristic oscillations form. These are signaled by topology violations. Although the question of measuring the topographic properties of the map is not trivial, cf. [3, 19, 18] we have found a simple criterion which proved reliable in practical applications. We consider the distance $\alpha = \|\mathbf{r}' - \mathbf{r}''\|$ which is the distance between the first and second closest neuron to the current data

[1] The normalization factor $\left(2\pi\sigma_{\mathbf{r}}^2\right)^{-\frac{D}{2}}$ was introduced in order that the average force exerted on $w_{\mathbf{r}}$ is independent on the local values of $\sigma$.

point where $= \alpha = 0, 1, \ldots$. If for any data point $\alpha > 1$ meaning that the first and second winner are not neighbors than there is a violation of topology in the region of the data point which means that $\sigma$ has fallen below its critical value. In other words $\alpha > 1$ signals the onset of the phase transition to the over-fitting (topology violating) regime.

Consequently, our approach consists in keeping $\sigma_{\mathbf{r}}$ fluctuating around its (unknown) critical value $\sigma_{\mathbf{r}}^{\mathrm{crit}}$, i.e. we decrement $\sigma_{\mathbf{r}}$ at each step as

$$\Delta\sigma_{\mathbf{r}} = -\frac{1}{NT_\sigma}\sigma_{\mathbf{r}} \quad \forall\mathbf{r} \tag{10}$$

and reset whenever $\alpha > 1$ the local values of $\sigma$ as

$$\sigma_{\mathbf{r}} := \max\left(\sigma_{\mathbf{r}}, \alpha\, \exp\left(-\frac{2\,(\mathbf{r}-\mathbf{R})^2}{\alpha^2}\right)\right), \quad \text{where } \mathbf{R} = \frac{1}{2}\left(\mathbf{r}' + \mathbf{r}''\right). \tag{11}$$

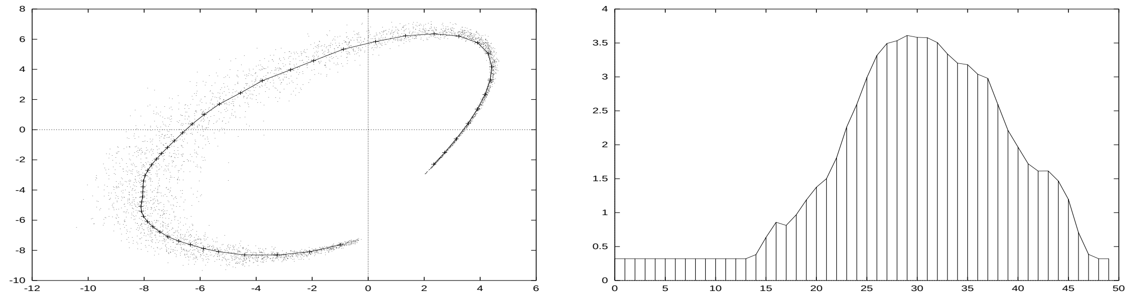

Figure 4: The map of a two-dimensional data distribution of varying scattering width onto a one-dimensional chain of 50 neurons (left). Final values of $\sigma_r$ along the neural chain produced by enforcing topology preservation (right).

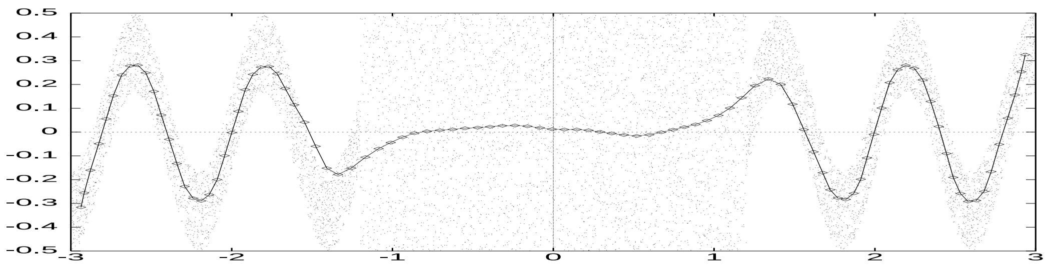

Figure 5: Mapping a noisy distorted *sin*-wave onto a chain of neurons. The noise is largely amplified and clipped in the center of the data distribution.

As a result, the map fluctuates around the PM due to the phase transition taking place each time the phase barrier corresponding critical value $\sigma_c$ is crossed. In order to average over the fluctuations each neuron keeps a second pointer $\overline{\mathbf{w}}_{\mathbf{r}}$ obtained by the moving average

$$\Delta \overline{\mathbf{w}}_{\mathbf{r}} = \frac{1}{KNT_\sigma} \left( \mathbf{w}_{\mathbf{r}} - \overline{\mathbf{w}}_{\mathbf{r}} \right) \tag{12}$$

over the fluctuations, where $K$ is of the order of 10. The $\overline{\mathbf{w}}_{\mathbf{r}}$ provide in most cases a very good first order data model. Further improvements depend on the task. In the case of modeling a functional relationship (see introduction) one may use the $\overline{\mathbf{w}}_{\mathbf{r}}$ to investigate the properties of the noise $\eta$ in order to improve the model. For the PM case, an essential improvement consists in using the $\overline{\mathbf{w}}_{\mathbf{r}}$ as starting positions for a final step in the sense of the iterative HS algorithm. This can be implemented more easily by monitoring directly the averages over the data in each domain. Hence, instead of $\overline{w}_{\mathbf{r}}$ each neuron gets a second pointer $\overline{v}_{\mathbf{r}}$ updated if the neuron is the winner as

$$\Delta \overline{v}_{\mathbf{s}} = \frac{1}{KT_\sigma} \left( \mathbf{v} - \overline{v}_{\mathbf{s}} \right) \tag{13}$$

The set $\{\overline{v}_{\mathbf{r}} \mid \mathbf{r} = 1, ..., N\}$ are the final result of the algorithm, i. e. they represent the PM in input space. Several toy applications of the present algorithm may be found in the Figures 4 and 5.

## 3.3 Sparse data sets

The above algorithm hinges on the abundance of data points which signal the folding via the topology violations. This may fail if the number of data points is small. For this case, a very sensitive criterion for the emergence of the critical fluctuations was found to be a wavelet transform [5, 4] of the map. For a one-dimensional SOM we use the Gabor transform

$$g_{r'} = \frac{1}{\sqrt{2\pi u_{r'}}} \left\| \sum_{k=1}^{N} w_{r'} \, \exp\left( \frac{-(k-r')^2}{2u_{r'}^2} \right) \exp\left(-i \; k\omega_{r'}\right) \right\| \tag{14}$$

where both the frequency $\omega_{r'}$ of the kernel and the width are functions of the current values of $\sigma_{r'}$ so that the kernel is always in resonance with potential foldings. At the critical point $\sigma = \sigma^{\mathrm{crit}}$ the wavelength of the emerging folds is $\lambda = 4.04\sigma l$, where $l$ is the average distance between the neurons in that region, cf. [15]. Choosing $\omega_{r'} = u_{r'} = 4\sigma_{r'}$ causes $g_{r'}$ to jump by an order of magnitude when $\sigma_{r'}$ drops below $\sigma_{r'}^{\mathrm{crit}}$. Hence, $g_{r'}$ is the desired sensitive criterion for detecting the onset of the phase transition.

## 3.4 Principal manifolds

So far, we have considered mainly the case of principal curves. However nothing in the algorithm presented above for the self–regulation of the smoothness parameter $\sigma$ is restricted to the case of a principal curve, i. e. a chain of neurons. We have applied the algorithm to several higher dimensional tasks, one toy example being given in Fig. 6.

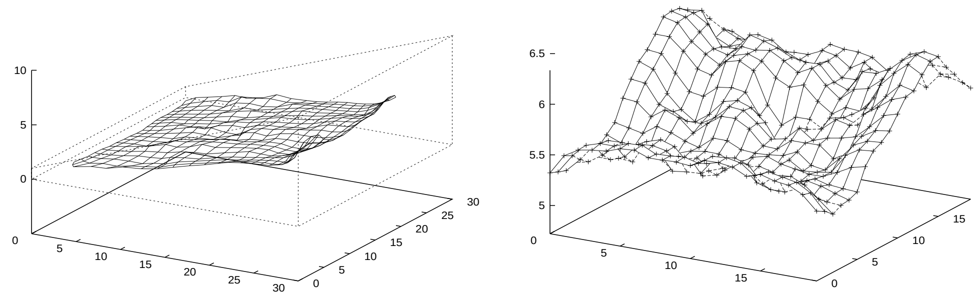


Figure 6: Embedding a two-dimensional lattice into a three-dimensional data set (left). Values of the neighborhood widths $\sigma_{\mathbf{r}}$ as a function of the lattice sites $\mathbf{r}$ (right).

# 4 Adaptive topologies

In important applications data sets are multiply connected. Here a generalization of the above definition lead to generalized principal curves, which are allowed to possess a number of branching points, but do observe — except in the vicinity of the branching points — the same conditions as PCs considered so far. Since it is hard both to specify an appropriate topology and to later match it to the data, we have chosen a different approach based on the neural gas algorithm. It allows to first represent the data structure by virtual positions of neurons and later to observe phase transitions very similar to that present in Kohonen maps. Thus, the adaptive-topology maps resemble locally the SOM-approach, but provides greater generality if necessary.

## 4.1 Generalized principal curves

We are now going to present an algorithm for the extraction of a nonlinear data model so that non- or multiply connected data structures can be represented without any prior knowledge of the number of components and their respective structure. We will consider pseudo one–dimensional data manifolds so that the task is to find generalized principal curves. The algorithm presented above rested on the self–organizing feature map which requires the topology of the net (lattice) to be specified beforehand. In the present section we describe an algorithm which finds the correct topology automatically.

## 4.2 The neural gas algorithm

An "intelligent" algorithm, instead of relying on a prespecified topology for the manifold in the output (neuron) space $\mathcal{A}$, must be capable to infer this topology from the data. The *neural gas* algorithm [12, 14, 13, 7] is suitable for this purpose while otherwise exhibiting similar properties as Kohonen's learning rule (see below). The only difference is in the definition of the neighborhood function which relies on the rank $R(r)$ in the ordered sequence of distances $\|w_r - v\|$. For the best matching unit $s_0$ we have

$R(s_0) = 0$, the second-best has $R(s_1) = 1$ and so on. Thus,

$$h_r(\mathbf{v}, \mathbf{w}) = \exp\left(-R(r)/\sigma\right) \tag{15}$$

and the weights are learned according to

$$\Delta \mathbf{w}_r = \epsilon h_{r'}(\mathbf{v}, \mathbf{w}_r)(\mathbf{v} - \mathbf{w}_r) \tag{16}$$

while decreasing $\sigma$. Eventually, the topology is represented in terms of a connectivity matrix arising from a simple Hebbian learning rule, cf. eq. (18) below.

However, the topology learned in this way is that of the noisy data and not of the generalized principal curve to be constructed. A principal curve is obtained from the *neural gas* by keeping the interaction width $\sigma$ sufficiently high so that the weights (virtual positions of the neurons) are forced into chain-like structures. The algorithm given below will self-consistently adapt these widths locally to obtain the desired generalized principal curve representation of the data set.

## 4.3 Phase transitions in the neural gas

Phase transitions to the over-fitting regime are observed also in the neural gas, although there is no fixed topology. The point is that for sufficiently large $\sigma$ the neurons are mapped chain-like into the rectangular input space. This is an immediate consequence of the cooperativity in learning introduced by the neighborhood function. What is the analog of the topology violations observed in the Kohonen map when $\sigma < \sigma_c$? What one observes is that in the over-fitting regime the first and second winner are no longer neighbors in input space. There are further neurons lying between the two in the sense that their (Euclidean) distance from both the first and the second winner is smaller than the distance between the first and second winner themselves. The number of these neurons "in between" may be used for fixing the local value of the neighborhood function optimally.

## 4.4 The algorithm

For each input signal $v$ drawn from a distribution $P(v)$ the first and second winner $s_0$, $s_1$ are calculated. The neighborhood parameter $\sigma$ is initialized at e. g. $N/3$ (with $N$ being the number of units) and is decreased slowly later on:

$$\sigma_r := \max\{\sigma_r - \varepsilon_\sigma \sigma_r, 1\}. \tag{17}$$

The strength $C_{rs}$ of the links develop by weakening in each step all links simultaneously, followed by a strengthening the link between the first two winners.

$$\Delta C_{ij} = -\frac{1}{N}\varepsilon_c C_{ij} \;\; \forall i, j \quad \text{and} \quad \Delta C_{s_0 s_1} = \varepsilon_c \tag{18}$$

If in this way a new link has been created (i. e. has grown across the threshold) the neighborhood parameter $\sigma_r$ is increased for all neurons with pointers (virtual positions)

$w_r$ between $w_{s_0}$ and $w_{s_1}$, counteracting to the persistent decrease of $\sigma$. We consider those neurons as situated between the first and second winner for which

$$\|w_r - w_{s_0}\| < \|w_{s_0} - w_{s_1}\| \quad \text{and} \quad \|w_r - w_{s_1}\| < \|w_{s_0} - w_{s_1}\|. \tag{19}$$

$\sigma$ is increased towards an estimated optimal $\hat{\sigma}_{\text{opt}}$ which is proportional to the number of neurons satisfying (19) with a global proportionality constant.

$$\Delta\sigma_r = \mu_\sigma(\hat{\sigma}_{\text{opt}} - \sigma_r) \text{ if } \sigma_r < \hat{\sigma}_{\text{opt}} \tag{20}$$

Further, the $\sigma_r$ must not exceed their initial value.

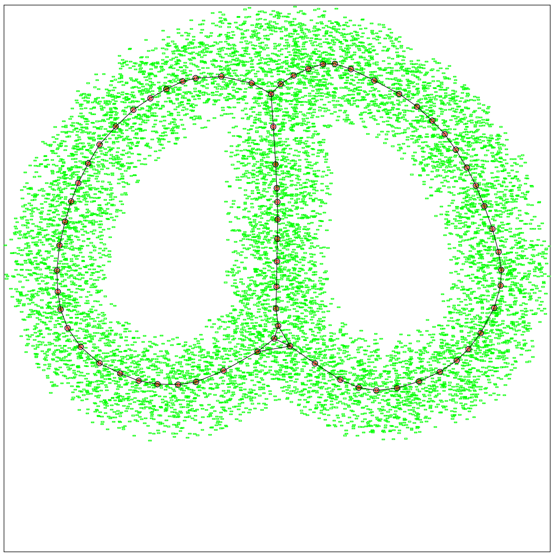

Figure 7: A multiply connected data set represented by a generalized principal curve, obtained with the algorithm described in Sec. 4.4. Smoothness of the principal curve was adapted locally by controlling the transition to the overfitting regime.

Next, sort the list of distances $\|w_r - w_{s_0}\|$ in an ascending order and define a modified neighborhood function $h_r(v, \{w\})$ by assigning ranks $R_r = 1$ to all units which are directly connected to the winning neuron, while using the usual ranks otherwise. The update rule of the algorithm finally reads

$$\Delta w_r = \varepsilon \exp\left(-\frac{2R_r^2}{(\sigma_r + \sigma_{s_0})^2}\right)(v - w_r)\,. \tag{21}$$

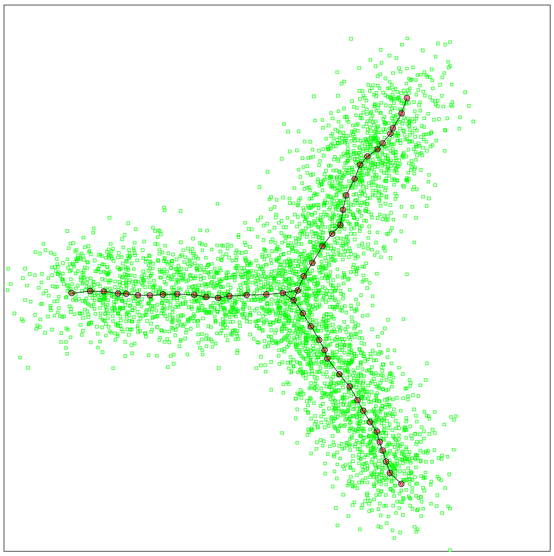

Figure 8: Same as Fig. 7 for a Y-shaped branching data distribution.

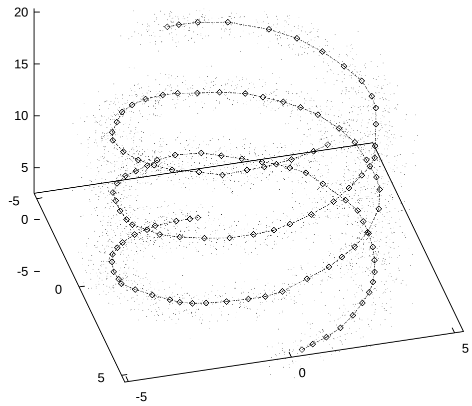


Figure 9: Solution of a two–spirals problem by the algorithm of the present paper. Note that the data manifold is disconnected and that the smoothness of the generalizxed principal curve is self-controlled.

This algorithm solves the principal curve problem in a quite general way, examples may be found in Figs. 7, 8, and 9 further details being given in Ref. [1]. Our algorithm combines learning at different levels, in particular of the network structure, while retaining the capabilities of the learning mechanisms at the other levels. The algorithm is relatively efficient since the most time consuming stage is the averaging process. Forthcoming work will address the case of generalized principal manifolds of dimension greater than one.

# 5 Revealing cluster structures in nonlinear data sets

The methods given above allow the construction of principal manifolds (PMs) in a reliable way. Once the PM is obtained, it may form the basis of further data analysis. One application is the visualization of the cluster structure of high dimensional nonlinear data sets. In fact, the projections of the data points on the PM faithfully reflect the cluster structure of the high dimensional data manifold. If the PM is one– or two–dimensional the cluster structure can be found by immediate visual inspection.

For a discussion we consider two examples. Fig. 10 presents a nonlinear clustered data set in three dimensional space. A set of real world data is shown in Fig. 11. In both cases the PM shown was constructed by means of Kohonen's algorithm. Obviously the projection of data points from the 11-dimensional input space onto the principal manifold (given explicitly in Fig. 11) reveals the cluster structure of the data.

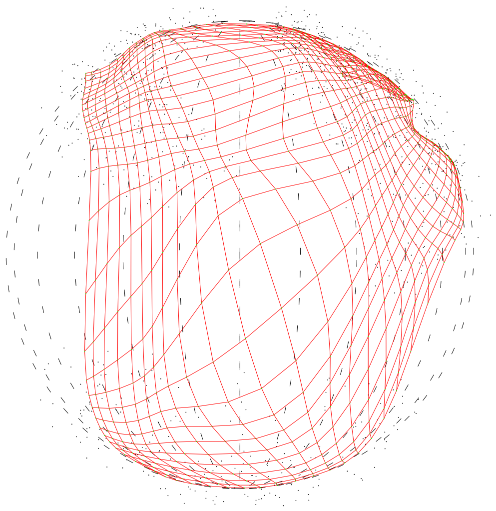

Figure 10: Principal manifold for a nonlinear set of data. The data points (dots) form three clusters, a square (top right), a circle (bottom), and an ellipse (right) mapped noisily onto a sphere (dashed).

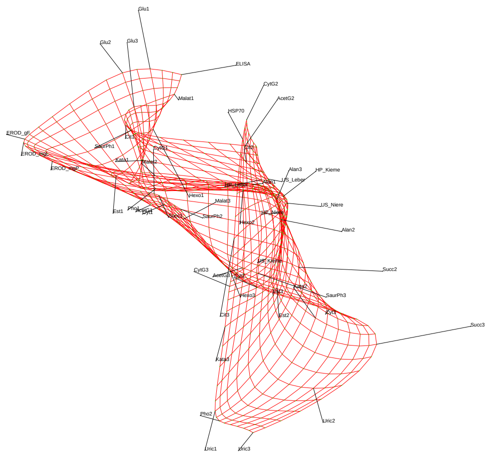

Figure 11: Principal manifold for a set of real world data and their projections. The data points are 11-dimensional vectors representing a biomarkers sampled both over time and over groups of individuals.

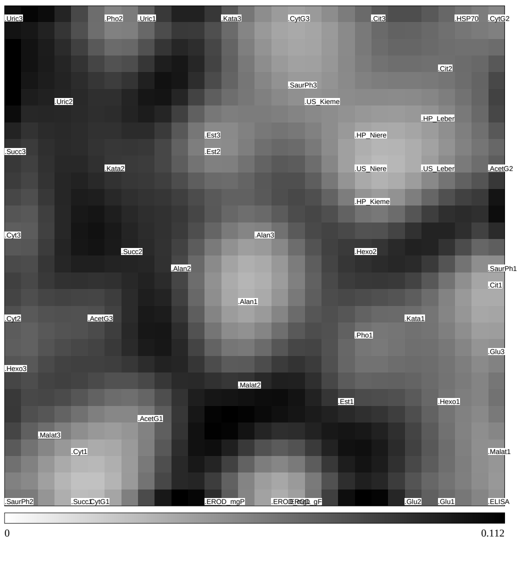


Figure 12: Map of the cluster structure of the real world data of Fig. 11. The gray values mark the average distance taken in the input space of a neuron to its neighbors (according to position in the lattice $\mathcal{A}$). Black corresponds to largest distances. Note that the distance between data points is given mainly by the blackness of barriers between the points.

However the Figures reveal also a serious drawback of this method arising from the density dependent resolution of the self–organizing map. This means that the concentration of neurons inside the clusters is high whereas the region between clusters is only scarcely populated. On the map this means that the regions between the clusters are shrinked whereas the clusters appear magnified. This is a counterintuitive representation of the cluster structure.

In order to overcome this drawback one may either use the methods given in [9, 2] for the control of the magnification exponent of the map or one tries to reintroduce the distance between the clusters into the representation. This can be done conveniently by a method given by Ultsch et al. [16, 17]. It essentially consists in displaying as a landscape the average distance taken in the input space of a neuron to its neighbors (according to position in the lattice $\mathcal{A}$).

Fig. 12 shows an application to a real world data set. One may clearly recognize the cluster structure by notifying the dark walls which mark the boundaries between clusters.

The real world data to some extent demonstrate the superiority of the nonlinear principal component analysis (NPCA) over the normal PCA. The latter yields three large eigenvalues for the data set considered namely $\lambda_1 = 10.04, \lambda_2 = 6.72, \lambda_3 = 4.10$ in the one case and $\lambda_1 = 9.05, \lambda_2 = 7.44, \lambda_3 = 6.77$ This suggests a three–dimensional space for the embedding of the data. The reason for this behavior becomes obvious from Figs. 13 and 14. The 2–dimensional principal manifold (PM), while still smooth is shaped roof like which is registered as a three–dimensional object by the PCA.

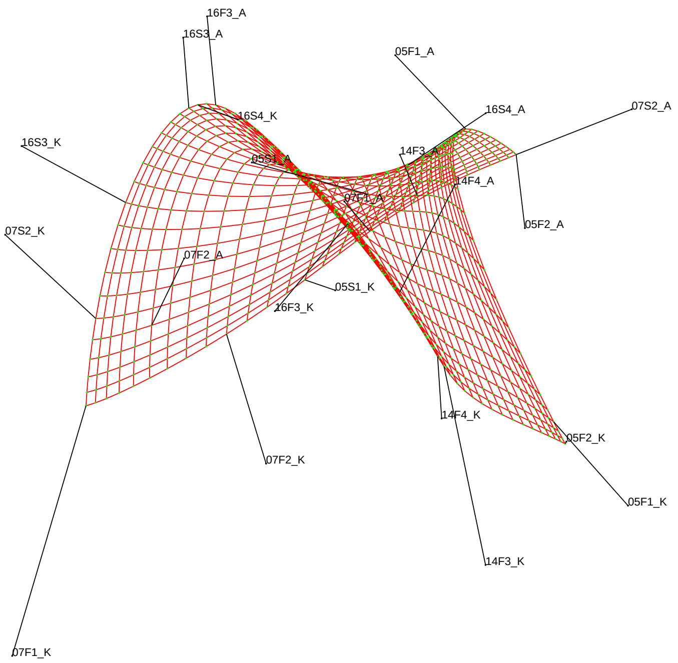


Figure 13: One set of biomarkers characterizing the Aich's and Koersch's population.

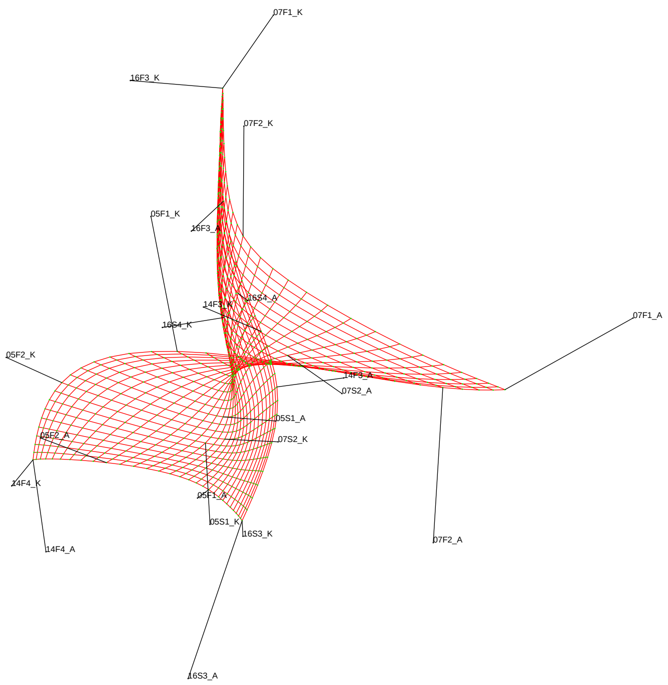


Figure 14: Another set of biomarkers characterizing the Aich's and Koersch's population.

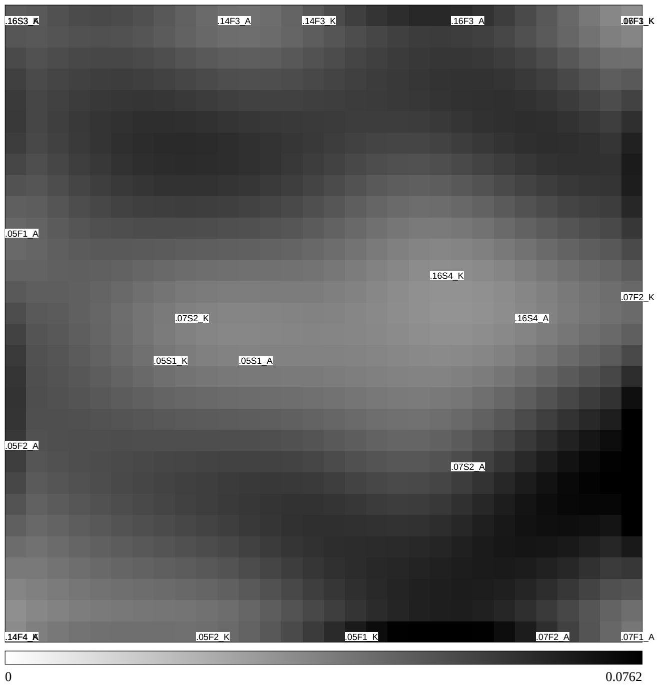


Figure 15: The grey plot for the data set of Fig. 14

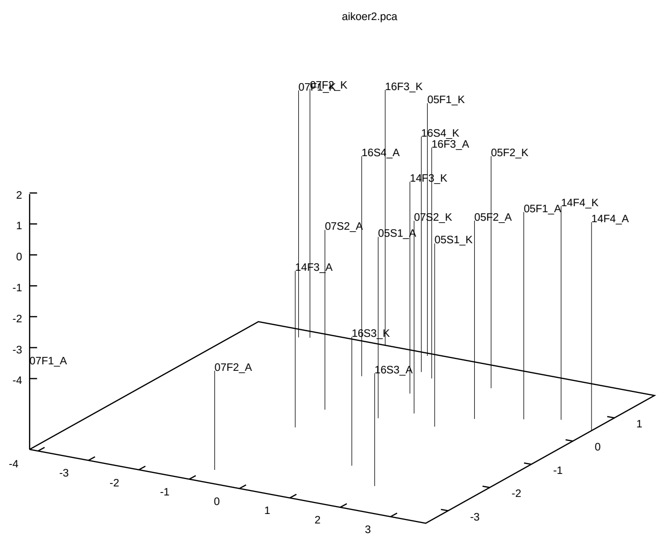


Figure 16: The PCA for the data set generating the NPCA of Fig. 14.

The difference becomes also prominent if we compare the clustering of the data as seen by both PCA and NPCA. We depict the data as projections into the two–dimensional space spanned by the first two eigenvectors, see Fig. 16 and as projections onto the two–dimensional PM, cf. Fig. 14, respectively. For a discussion we compare the cluster structures. There is a correspondence for the two data points $16S3\_A$ and $16S3\_K$ which form a cluster in both the PCA and NPCA analysis. However the data points $05S1\_A$, $05S1\_K$, and $07S2\_K$ cluster according to the NPCA but not in the PCA representation. On the other hand the cluster consisting of $16F3\_A$, $16S4\_K$, and $05F1\_K$ form a cluster in the PCA representation whereas in the NPCA they are widely separated. In view of the large errors in the representation by the first two PCA components it is not too surprising that the linear cluster analysis fails. We have observed better agreement between the linear and nonlinear analysis for other cases where the error in the PCA representation is not so large.

The results obtained so far are encouraging. However there are still some obvious shortcomings of the method. Above all this concerns the treatment of the data points populating the boundaries of the data set. Both Kohonen's and the neural gas algorithm tend to concentrate the lattice in the inner regions of the data cloud. This is clearly seen in Figs. 10 and 11 as well as in 8. The problem is considered in [11, 10] but so far there is no feasible tool for circumventing this problem in the general case.

# 6 Concluding remarks

We developed two new algorithms for the general task of extracting nonlinear principal manifolds from high–dimensional and noisy data sets. We have demonstrated that the method is applciable to real world data sets and that in the case of nonlinear data manifolds the method allows to extract informations not visible from the conventional linear methods of data analysis. The self-regulation of the parameters of the algorithm is a major step towards the establishment of the method as a general tool of nonlinear data analysis.

ACKNOWLEDGMENT: The work was supported by a grant of the Umweltforschungszentrum Leipzig/Halle GmbH. During the project we largely benefitted from the cooperation with M. Herrmann (Göttingen) and with H.-U. Bauer (Göttingen) and M. Welk (Leipzig). Numerous discussions and important contributions to the results reported above are gratefully acknowledged.

# References

[1] G. Balzuweit, R. Der, M. Herrmann, and M. Welk. An algorithm for generalized principal curves with adaptive topology in complex data sets. Technical Report 3/97, Institut für Informatik, Universität Leipzig, 1997. URL: http://www.informatik.uni-leipzig.de/ der/Veroeff/gen.pc.ps.gz.

[2] H.-U. Bauer, R. Der, and M. Herrmann. Controlling the magnification factor of self-organizing feature maps. *Neural Computation*, 8(4):757–771, 1996.

[3] H.-U. Bauer and K. R. Pawelzik. Quantifying the neighborhood preservation of Self-Organizing Feature Maps. *IEEE Trans. on Neural Networks*, 3(4):570–579, 1992.

[4] C. K. Chui. *An Introduction to Wavelets*, volume 1 of *Wavelet Analysis and its Applications.* Academic Press, Inc., 1992.

[5] I. Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[6] R. Der and M. Herrmann. Critical phenomena in self-organized feature maps: A ginzburg-landau approach. *Phys. Rev. E*, 49(5):5840–5848, 1994.

[7] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 625–632. The MIT Press, 1995.

[8] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[9] M. Herrmann, H.-U. Bauer, and R. Der. "optimal magnification factors in self-organizing feature maps". In *Proc. ICANN'95*, volume 1, pages 75–80, Paris, 1995. EC2 & Cie.

[10] V. M. M. Hulle. Erratum. *Neural Networks*, 10(6):1165–1166, 1997.

[11] V. M. M. Hulle. Topology-preserving map formation achieved with a purely local unsupervised competitive learning rule. *Neural Networks*, 10(3):431–446, 1997.

[12] T. Martinetz and K. Schulten. A "Neural-Gas" network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Proc. Int. Conf. on Artificial Neural Networks* (Espoo, Finland), volume I, pages 397–402, Amsterdam, Netherlands, 1991. North-Holland.

[13] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7(2), 1994.

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

[15] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps: An Introduction.* Addison-Wesley, Reading, MA, 1992.

[16] A. Ultsch. Knowledge extraction from self-organizing neural networks. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification*, pages 301–306, Berlin, 1993. Springer.

[17] A. Ultsch. Self-organizing neural networks for visualization and classification. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification*, pages 307–313, Berlin, 1993. Springer.

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

[19] T. Villmann, R. Der, and T. Martinetz. A novel approach to measure the topology preservation of feature maps. In M. Marinaro and P. G. Morasso, editors, *Proc. ICANN'94, Int. Conf. on Artificial Neural Networks*, volume I, pages 298–301, London, UK, 1994. Springer.