Building Nonlinear Data Models with Self–Organizing Maps

Ralf Der[1], Gerd Balzuweit[1], Michael Herrmann[2]

[1] University of Leipzig, Institute of Informatics
P. O. Box 920, 04109 Leipzig, Germany
[2] RIKEN, Lab. of Information Representation
2-1 Hirosawa, Wako-shi, 351-01 Saitama, Japan

Abstract. We study the extraction of nonlinear data models in high dimensional spaces with modified self-organizing maps. Our algorithm maps lower dimensional lattice into a high dimensional space without topology violations by tuning the neighborhood widths locally. 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. The performance of the algorithm is demonstrated for one- and two-dimensional principal manifolds and for sparse data sets.

1 Introduction

Artificial neural networks provide convenient tools for reconstructing nonparametric data models from noisy data. Consider data representing a functional relation $y = f(x)$ corrupted by noise, i.e. the observations are given by $y = f(x) + \eta$, where η is the noise which may or may not depend on x. Building a data model means extracting the systematic term $f(x)$ from the noisy data. If the noise also affects the x values the problem transforms into the construction of principal manifolds (PMs) that model data distributions by providing a curvilinear coordinate system for a dimension reduced representation of the data. PMs, principal curves e.g., may be defined self-consistently by the requirement that each point on the PM is the average of the data points projecting to it, cf. [1]. Thus, the mean square deviations (MSD) of the data from the PM are minimized locally at each base point. Averaged over the input distribution, however, the PM is only a stationary point with respect to the MSD [1], i.e. for some directions in the function space the PM belongs to, it is stable, for others unstable. So far there exist no explicit criteria for the existence, uniqueness and stability of PMs. It is known from the simulation of many data sets [1] that stability is guaranteed by local averaging the span of average being guided *globally* by cross validation [1].

Kohonen's algorithm provides a nonlinear mapping of some lattice (representing the physical positions of neurons) into the data manifold in input space. Thus, a piecewise linear approximation of the desired curvilinear coordinate system is established (cf. [3, 4]). Whether or not these coordinates reflect the main features of the data or, more precisely, whether they are principal curves of the

data becomes in the context of self-organizing maps (SOMs) a stability problem. For this problem – and in accordance with the empirical observations for principal manifolds – it could be shown [3, 5] that stability is ensured by a sufficient range of local collaboration, i.e. by local averaging over a sufficiently wide span. This indicates the principal manifold are saddle points with respect to the MSD which are stable with respect to long ranged variations and unstable for short ranged variations. Averaging or neighborhood interaction stabilizes PMs because the short ranged deviations are suppressed. In [3, 4] stability problems have not been addressed, i.e. the neighborhood parameter has been assumed to vary only in the stable range. Since on the other hand the averaging disrupts the representation of specific data features, and thus tends to increase the MSD, the neighborhood coherence should not be to long-ranged.

The present paper focuses on a choice of this parameter that compromises between the requirements of smoothness and small MSD error. We start from Kohonen's unsupervised learning rule cf. [2] to develop a simple and robust algorithm for learning a good approximation of the principal manifold. As compared to the HS algorithm, it has the advantages of being on-line learning, not dimension specific and uses a *locally* defined width for the smoothing operation. This is important for data distributions with locally varying variance of the noise (multiplicative noise). Most importantly these widths are self-regulating. Our approach rests on a new principle which exploits the specific dynamical properties of the first-order phase transition [3, 5] induced by the noise. The approach is shown to work also for sparse data sets and should therefore be favorable also in the case of high-dimensional inputs.

2 Maps with locally adaptive smoothness

Let us assume that the data embedded in some d-dimensional space $\mathcal{X}$ actually are 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 points $\mathbf{r} \in [1, N]^D$ may be considered as the physical positions of N^D neurons. Upon presentation of a data vector $\mathbf{x}$, the learning step for the synaptic vectors $\mathbf{w_r}$ is

$$\Delta \mathbf{w_r} = \varepsilon \, h_{\mathbf{r},\mathbf{r}'} \, (\mathbf{x} - \mathbf{w_r}) \,, \tag{1}$$

where the position $\mathbf{r}'$ of the winner (best matching) neuron is determined by $\mathbf{r}' = \arg\max_{\mathbf{r}} \|\mathbf{x} - \mathbf{w_r}\|$. The neighborhood function $h_{\mathbf{r},\mathbf{r}'} = \left(2\pi\sigma^2\right)^{-\frac{D}{2}} \exp\left(-(\mathbf{r} - \mathbf{r}')^2/(2\sigma^2)\right)$ defines the range of cooperativity between neurons which controls the smoothness of the map: Its radius of curvature ρ obeys $\rho > 2d$ where d is about the Euclidean distance between neurons in input space which are a distance σ apart on the lattice. (The normalization factor $\left(2\pi\sigma^2\right)^{-\frac{D}{2}}$ was introduced in order that the average force exerted on $w_{\mathbf{r}}$ is independent on σ). For varying width of the data scattering we need the smoothness

of the map to be defined locally. This may be achieved by an individual neighborhood $\sigma_{\mathbf{r}}$ for each neuron, so that

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

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

The crucial point now is the determination of the local values $\sigma_{\mathbf{r}}$. The principal manifold can be mapped topographically onto the lattice because of the dimensions match by definition. However, with the data points scattering about the PM we have to compromise between two options. On the one hand the lattice should be mapped tightly to the PM which requires a small σ (curvature) if the PM is nonlinear. On the other hand the stiffness and hence σ should be sufficiently large in order to avoid the folding due to the dimensional mismatch.

For the definition of the optimal $\sigma_{\mathbf{r}}$ we exploit the dynamics of the phase transitions induced by the scattered data points. Namely, if σ is lowered below a value σ^{crit} the representation of the main data feature by the map gets distorted by other 'noisy' features. The transition has has been shown earlier [5] to proceed through a either of two phases, one preserving and one violating topology. Since the latter is more pronounced, emerging folding into secondary features will be signaled immediately by topology violations. Concentrating on the first and second winner the criterion for the occurrence of a fold is $\alpha > 1$ where $\alpha = \|\mathbf{r}' - \mathbf{r}''\|$ is the distance between the first and second winner in the lattice.

In contrast to [6], where the neighborhood width has been updated following an energy function resulting in the occurrence of local minima and a slow convergence, our approach here consists in *keeping* $\sigma_{\mathbf{r}}$ *fluctuating* around its critical value $\sigma_{\mathbf{r}}^{\text{crit}}$. The result of the algorithm is given in terms of sliding averages over the fluctuations rather than a convergent network state. For this purpose we decrement $\sigma_{\mathbf{r}}$ at each step as

$$\Delta\sigma_{\mathbf{r}} = -\frac{1}{NT_\sigma}\sigma_{\mathbf{r}} \quad \forall \mathbf{r} \tag{3}$$

and reset whenever $\alpha > 1$ the σ's in the vicinity of the topology distortion as

$$\sigma_{\mathbf{r}} := \max\left(\sigma_{\mathbf{r}}, \alpha \ \exp\left(-\frac{2\,(\mathbf{r}-\mathbf{R})^2}{\alpha^2}\right)\right), \quad \text{where } \mathbf{R} = \frac{1}{2}\,(\mathbf{r}' + \mathbf{r}'')\,. \tag{4}$$

As a result, the map fluctuates around the PM due to the phase transition taking place each time the critical value σ^{crit} 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

$$\Delta\overline{\mathbf{w}}_{\mathbf{r}} = \frac{1}{KNT_\sigma}\,(\mathbf{w}_{\mathbf{r}} - \overline{\mathbf{w}}_{\mathbf{r}}) \tag{5}$$

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 modelling a functional relationship (see introduction) one

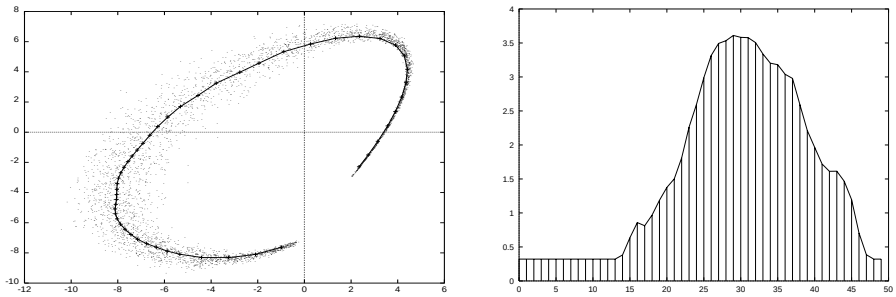

Fig. 1. The map of a two-dimensional data distribution of varying scattering width onto a one-dimensional chain of 50 neurons (left). Final values of σ_r along the neural chain produced by enforcing topology preservation (right).

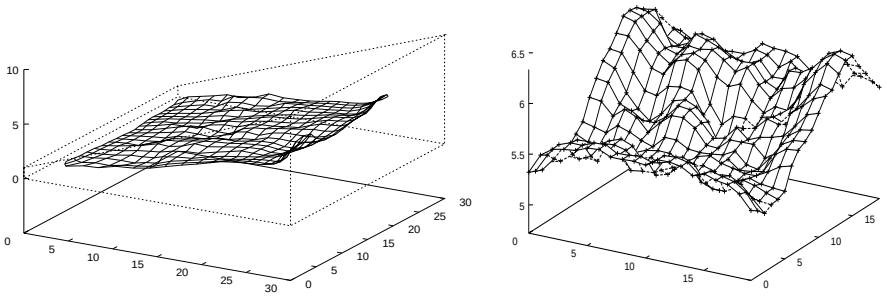

Fig. 2. Embedding a two-dimensional neural lattice in a three-dimensional data set (left). Values of the neighborhood widths $\sigma_{\mathbf{r}}$ as a function of the net position $\mathbf{r}$ (right).

may use the $\overline{\mathbf{w}}_{\mathbf{r}}$ to investigate the properties of the noise η 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 $\overline{v}_{\mathbf{r}}$ measuring the local average of the input data which may deviate from $\overline{\mathbf{w}}_{\mathbf{r}}$ if the principal manifold is curved. $\overline{v}_{\mathbf{r}}$ is updated if neuron $\mathbf{r}$ is the winner as $\Delta\overline{v}_{\mathbf{s}} = \frac{1}{KT_\sigma}(\mathbf{v} - \overline{v}_{\mathbf{s}})$ 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.

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 [7] of the map. For a one-dimensional SOM we use the Gabor transform

$$g_{r'} = \frac{1}{\sqrt{2\pi}u_{r'}} \left\| \sum_{k=1}^{N} w_{r'} \exp\left(\frac{-(k-r')^2}{2u_{r'}^2}\right) \exp\left(-i\, k\omega_{r'}\right) \right\| \qquad (6)$$

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

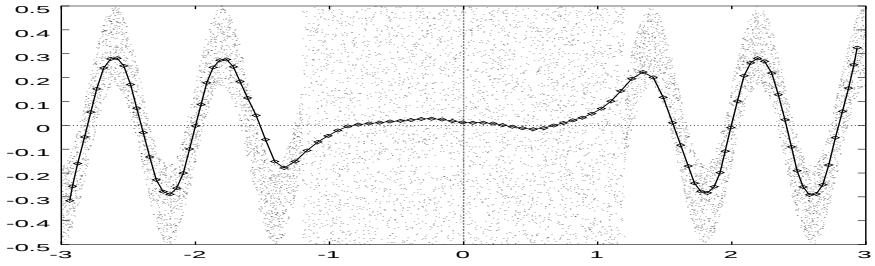

Fig. 3. Mapping a noisy *sin*-wave onto a chain of neurons. The strong noise in the center of the data distribution is clipped by the map.

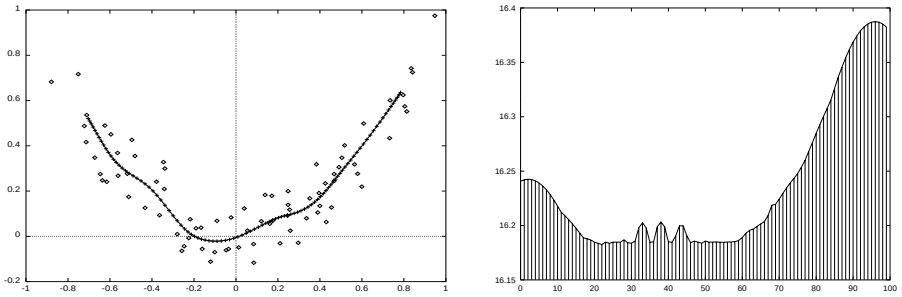

Fig. 4. Using wavelet transform to control the neighborhood widths in a sparse data set. Display of the emerged map (left) and σ_r-values (right).

foldings. At the critical point $\sigma = \sigma^{\rm crit}$ 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. [3]. Choosing $\omega_{r'} = u_{r'} = 4\sigma_{r'}$ causes $g_{r'}$ to jump by an order of magnitude when $\sigma_{r'}$ drops below $\sigma_{r'}^{\rm crit}$. Hence, $g_{r'}$ is the desired sensitive criterion for detecting the onset of the phase transition. In the algorithm we use (3) as before. If for the winner $g_{r'}$ exceeds a small threshold we use $\alpha = \kappa\sigma_{r'}$ in (4), observing $1 \leq \alpha \leq \sigma_{\rm max}$, where $\kappa = 1.2$ is an empirical factor. In the simulations a control of κ is obtained from monitoring the fluctuations of σ which optimally should stay in the region of a few percent.

4 Numerical simulations

We have applied the above algorithms to map both one- and two-dimensional lattices into higher-dimensional input spaces with inhomogeneous data distributions of effective dimension $D = 1$ or $D = 2$, respectively. The local scattering of the data points around the central manifold varied by up to an order of magnitude. In all simulations the algorithms were stable and produced the desired results. In particular the cooling time T_σ could be varied by more than an order of magnitude without instability problem. Parameters used in the simulations were $T_\sigma = 1500$, $\sigma_{\rm max} = N/3$, and $\varepsilon = 0.15$ (kept constant during the simulations. It should be noted that such relatively high values of ε allow for fast convergence and improve the efficiency of the average (5)), which in turn compensates the fluctuation in the final map $\overline{\mathbf{w}}_{\mathbf{r}}$. The first and the second example (Figs. 1 and 2, respectively) are carried out by the basic algorithm explained in

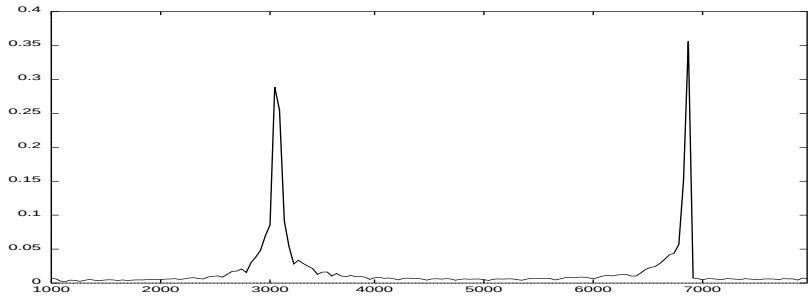

Fig. 5. Time course of the wavelet transform of one of the neurons during 7000 steps.

the second section. In the third example shown in fig. 4 we use wavelet transform to adjust the σ_r. A data set of 80 points and a chain of 100 neurons are chosen. In practical computation it is easy to use FFT for computing the Gabor transform function.

5 Conclusion

The present contribution is dedicated to the problem of building nonlinear data models with a modified Kohonen learning rule which learns its own parameters. In particular, the neighborhood range of the output units, which determines the smoothness of the produced maps, is an essential prerequisite for the formation of principal manifolds in the input data set. Our algorithm solves the task of the determination of the neighborhood parameter stably and in a local manner. This is of importance for processing sparse data sets with spatial heterogeneities. Therefore, the algorithm appears to be well suited for more complex tasks. In particular, the algorithm was found to be also stable with higher-dimensional (up to $d = 8$) inputs. The application to real world data is underway.

ACKNOWLEDGEMENT: One of the authors (RD) gratefully acknowledges the hospitality of RIKEN (Tokyo) he received during a visit in February 1996.

References

1. T. Hastie, W. Stuetzle, Principal curves. Journal of the American Statistical Association 84, 502–516 (1989)
2. T. Kohonen, *The Self-Organizing Map*, Springer-Verlag, 1995.
3. H. Ritter, T. Martinetz, K. Schulten, *Neural Computation and Self-Organizing Maps*, Addison-Wesley, 1992.
4. F. Mulier, V. Cherkassky, Self-Organization as an Iterative Kernel Smoothing Process. *Neural Computation* **7**, 1165-1177, 1995.
5. R. Der, M. Herrmann, Critical Phenomena in Self-Organized Feature Maps: A Ginzburg-Landau Approach, *Phys. Rev.* **E 49**:6, 5840–5848, 1994.
6. M. Herrmann, "Self-Organizing Feature Maps with Self-Organizing Neighborhood Widths", *Proc. 1995 IEEE Intern. Conf. on Neur. Networks*, 2998–3003, 1995.
7. C. K. Chui, *An Introduction to Wavelets*, Academic Press, 1992.