Self-organization and nonparametric regression

Tom Heskes and Bert Kappen

RWCP[1] Novel Functions SNN[2] Laboratory
Department of Medical Physics and Biophysics
University of Nijmegen, Geert Grooteplein 21
6525 EZ Nijmegen, The Netherlands

Submitted to ICANN'95

Abstract

We describe a framework for self-organization which relates the formation of topologic maps to minimization of a free energy function. In the zero noise limit the resulting on-line learning rule is similar to the Kohonen learning rule. We derive a fast EM-algorithm for finite training sets. Choosing different noise parameters for input and output variables we obtain an algorithm for knot placement in nonparametric regression. This algorithm naturally fits into projection pursuit regression when we treat the noise parameter as a projection vector.

1 Self-organizing maps

1.1 Introduction

Self-organizing maps have been used for many applications, including signal compression, combinatorial optimization, robot control, regression analysis and so on. Most of these methods rely on the well-known Kohonen learning rule [1]. In this paper we will describe a sligthly different version of the Kohonen learning rule, which has the advantage that it can be derived from a (free) energy function (see also [2, 3]). This energy function suggests an EM-algorithm that can be used in batch-mode. Furthermore, our model includes a noise parameter similar to the inverse temperature in statistical physics. Specific choices for this noise parameter link self-organizing maps to approaches for solving traveling salesman problems and to algorithms for nonparametric regression.

1.2 Free energy function

Suppose we have a network of n units ("neurons") labeled $1, \ldots, i, \ldots, n$. To each unit we assign a "local error" $e_i(W, \vec{z})$ which depends on the set of $N = m \times n$ adjustable parameters W and on the particular m-dimensional input vector $\vec{z}$. The probability that unit i "wins" on the input vector $\vec{z}$ is defined as

$$p_i(W, \vec{z}) \;=\; \frac{\mathrm{e}^{-\beta e_i(W,\vec{z})}}{Z(W,\vec{z})} \;\text{ with partition function }\; Z(W,\vec{z}) \;=\; \sum_{i=1}^{n} \mathrm{e}^{-\beta e_i(W,\vec{z})} \,,$$

with β a noise parameter. The free energy averaged over all input vectors $\vec{z}$ with probability density function $\rho(\vec{z})$ is

$$E(W) \;=\; -\frac{1}{\beta} \int d^m z \; \rho(\vec{z}) \; \ln Z(W, \vec{z}) \,, \tag{1}$$

[1]Real World Computing Program
[2]Dutch Foundation for Neural Networks

The formation of topological maps can be studied by choosing local errors

$$e_i(W,\vec{z}) = \frac{1}{2}\sum_{j=1}^{n} h_{ij}(\vec{z}-\vec{w}_j)^2 \,. \tag{2}$$

Here $\vec{w}_j$ is the m-dimensional weight vector of unit j. The $n \times n$ matrix H with components h_{ij} is called the lateral-interaction matrix. It contains the physical structure of the topological map. Usually the lateral-interaction strength is a monotonically decreasing function of the "hardware" distance between units in the topological map with typical range σ. Without any lateral interactions, i.e. for $\sigma = 0$, we obtain the clustering algorithm proposed in [4] (see also [5]). In the noiseless limit $\beta \to 0$, the energy function (1) converges to the "minimum distortion length" [3] which is based on a general interpretation of the Kohonen learning rule in the framework of signal transmission with noise (see also [2]).

1.3 Learning rules

Given a particular training pattern $\vec{z}$, drawn at random according to some probability density function $\rho(\vec{z})$, we apply the on-line learning rule

$$\Delta W = \eta \nabla_W \left[\frac{1}{\beta}\ln Z(W,\vec{z})\right] \quad \text{and thus} \quad \Delta\vec{w}_i = \eta\sum_{j=1}^{n} h_{ij}p_j(W,\vec{z})(\vec{z}-\vec{w}_j)\,.$$

performing (stochastic) gradient descent on the energy function (1). In the limit $\beta \to \infty$ there is one winner:

$$\kappa(\vec{z}) = \underset{k}{\operatorname{argmin}}\; e_k(\vec{z})\,.$$

The learning rule in the limit $\beta \to \infty$ is similar to the Kohonen learning rule [1], the difference being that in our model the determination of the winner is based on the local error $e_k(\vec{z})$ as in (2) instead of on the Euclidian distance $(\vec{z}-\vec{w}_k)^2$.

For a finite training set of P patterns $\vec{z}^{\mu}$ we can derive a faster learning rule based on the EM-algorithm [6]. The expectation step computes the function

$$Q(W|W') = \sum_{i=1}^{n}\sum_{\mu=1}^{P} p_i(W',\vec{z}^{\mu})e_i(W,\vec{z}^{\mu})$$

of the new weights W given the old weights W'. The maximization step yields the weights W that maximize this function $Q(W|W')$:

$$\nabla_{\vec{w}_i} Q(W|W') = \sum_{j=1}^{n}\sum_{\mu=1}^{P} p_j(W',\vec{z}^{\mu})h_{ji}(\vec{z}^{\mu}-\vec{w}_i) = 0 \quad \forall_i\,.$$

The solution to this set of equations is the closed-form expression

$$\vec{w}_i = \frac{\sum_{\mu=1}^{P}\sum_{j=1}^{n} h_{ji}p_j(W',\vec{z}^{\mu})\vec{z}^{\mu}}{\sum_{\mu=1}^{P}\sum_{j=1}^{n} h_{ji}p_j(W',\vec{z}^{\mu})}\,.$$

Since it can be shown that $E(W) \leq E(W')$ with equality iff $W = W'$ is a (local) minimum of function $E(W)$, this EM-algorithm is guaranteed to lead to a minimum of the energy function (1).

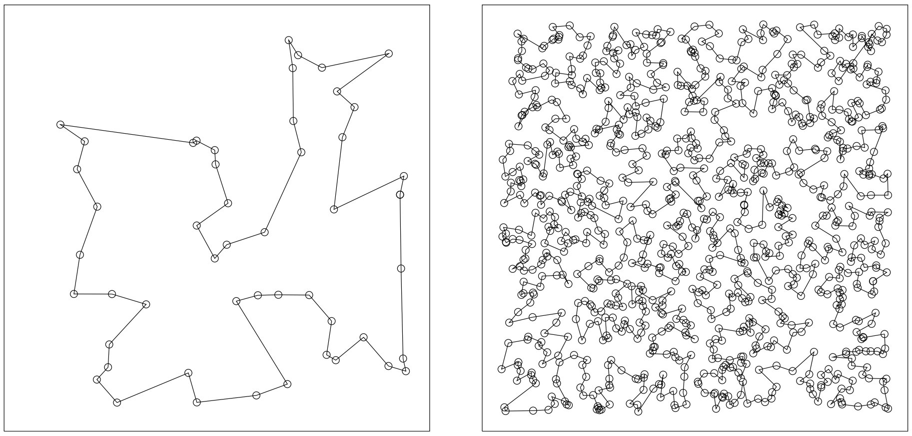

Figure 1: Solution of a traveling salesman problem on 50 cities with 100 units (left) and on 1000 cities with 2000 units (right).

1.4 Traveling salesman problem

Many methods based on the original Kohonen learning rule or the elastic-net algorithm have been proposed to solve traveling salesman problems (see e.g. [7]). An advantage of the elastic-net approaches is the possibility to use EM in conjunction with annealing. However, Kohonen networks in general yield better results, probably because the lateral interactions are not limited to nearest neighbors as for the elastic-net approaches. Our model shows that it is possible to use an EM-algorithm, while maintaining the flexibility of the lateral interactions. We illustrate this on traveling salesman problems in two dimensions with 50 cities and 1000 cities, using a ring of $n = 100$ and $n = 2000$ units, respectively. The lateral interaction matrix is of the form

$$h_{ij}(t) \;=\; \exp\left[-\frac{1-\cos(2\pi(i-j)/n)}{(\sigma(t)/n)^2}\right] .$$

Both the temperature $1/\beta$ and the interaction strength σ are gradually annealed to zero. The final solutions displayed in Figure 1 are obtained in 120 and 643 learning steps. Here these simulations serve as an illustration; a fair comparison with other approaches is outside the scope of this paper.

2 Nonparametric regression analysis

2.1 Knot placement for regression

In a regression problem we have a set of possibly inaccurate data points consisting of an M-dimensional inputs and a one-dimensional output:

$$y^\mu \;=\; f(\vec{x}^\mu) \;+\; \text{noise} .$$

In the combined $M+1$-dimensional space of inputs and output, we are looking for a smooth curve through the data. A common approach in nonparametric regression is to allow the function $f(\vec{x})$ to be a piecewise linear or maybe spline function with the different pieces joining at so-called knots. The role of the knots can be played by units in self-organizing feature maps. In fact, this is exactly what happens in the earlier stages of the traveling salesman problem where the units, still having nonzero lateral interaction, form a smooth ring roughly connecting the cities. In regression, however, there is the important restriction that just one output value is assigned to each input vector. When straightforwardly applied to regression problems, Kohonen-like algorithms may produce nonfunctional mappings [8]. To solve this problem we choose different noise parameters β_x and β_y for the input and the output, with $\beta_y \ll \beta_x \equiv \beta$, reflecting the larger uncertainty in the outputs if compared with the inputs. With notation $\vec{w}_i \equiv (\vec{u}_i, v_i)$ the local errors obey

$$e_i(W, \vec{z}) \;=\; \frac{1}{2} \sum_{j=1}^{n} h_{ij} \left[(\vec{x} - \vec{u}_j)^2 + \epsilon (y - v_j)^2 \right] \,, \tag{3}$$

with $\epsilon = \beta_y / \beta_x \ll 1$. In lowest order in ϵ we obtain the energy function

$$E(W) \;=\; E(U) \;+\; \frac{\epsilon}{2} \sum_{\mu=1}^{P} \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij} p_i^{\mu}(U) (y^{\mu} - v_j)^2 \,,$$

i.e. the energy function splits up in two independent parts. Minimization of the first term $E(U)$ yields a topological map of the input space. The resulting "gating factors" $p_i^{\mu}(U)$ pop up in the second term to "smooth" the function values. This algorithm for knot placement differs from the one proposed in [8] in a few important aspects: it is based on an energy function, it includes one extra noise parameter, and it can be trained with EM.

2.2 Projection pursuit regression

A disadvantage of the algorithm described in the previous paragraph is the prohibitive amount of knots needed for function approximation in many dimensions. An alternative approach is projection pursuit regression where the function $f(\vec{x})$ is approximated by a sum of one-dimensional functions (see e.g. [9, 10] for more extensive explanations of projection pursuit learning):

$$f(\vec{x}) \;=\; \sum_{\gamma=1}^{\Gamma} f_{\gamma}(\vec{a}_{\gamma} \cdot \vec{x}) \,,$$

with $\vec{a}_{\gamma}$ unit vectors. The functions $f_{\gamma}(\cdot)$ and projections $\vec{a}_{\gamma}$ are estimated sequentially to minimize the mean-squared distance with the residuals, i.e. for increasing γ we search for functions $f_{\gamma}(\cdot)$ and projections $\vec{a}_{\gamma}$ minimizing

$$L(\vec{a}_{\gamma}) \;=\; \frac{1}{2P} \sum_{\mu=1}^{P} \left[f_{\gamma}(\vec{a}_{\gamma} \cdot \vec{x}^{\mu}) - \left\{ y^{\mu} - \sum_{\gamma'=1}^{\gamma-1} f_{\gamma'}(\vec{a}_{\gamma'} \cdot \vec{x}^{\mu}) \right\} \right]^2 \,. \tag{4}$$

For each γ the projections $\vec{a}_{\gamma}$ and functions $f(\cdot)$ are estimated cyclically. Given a projection $\vec{a}_{\gamma}$ we are left with a one-dimensional nonlinear regression problem for which we can use the algorithm described in the previous section. The projection on $\vec{a}_{\gamma}$ comes in by changing the local error (3) into

$$e_i(W, \vec{z}, \vec{a}_{\gamma}) \;=\; \frac{1}{2} \sum_{j=1}^{n} h_{ij} \left[\{ (\vec{x} - \vec{u}_j) \cdot \vec{a}_{\gamma} \}^2 + \epsilon (y - v_j)^2 \right] \,.$$

The weights W and thus the knot positions then follow from the EM-algorithm minimizing the energy function (1). Spline interpolation between these knots fixes the nonparametric function $f_{\gamma}(\vec{a}_{\gamma} \cdot \vec{x})$. The Gauss-Newton method (see e.g. [10]) on the error (4) can be used to obtain a better estimate for the projection $\vec{a}_{\gamma}$, leading to a new function $f_{\gamma}(\cdot)$ and so on. The noise parameter β and interaction strength σ should be determined through cross-validation.

As an illustration we apply our algorithm on a training set consisting of $P = 50$ combinations $(\vec{x}^{\mu}, y^{\mu})$ with $\vec{x}$ homogeneously drawn from $[0, 1]^4$, $f(\vec{x}) = \sin(2\pi x_1 x_2)$ (x_3 and x_4 are "decoy

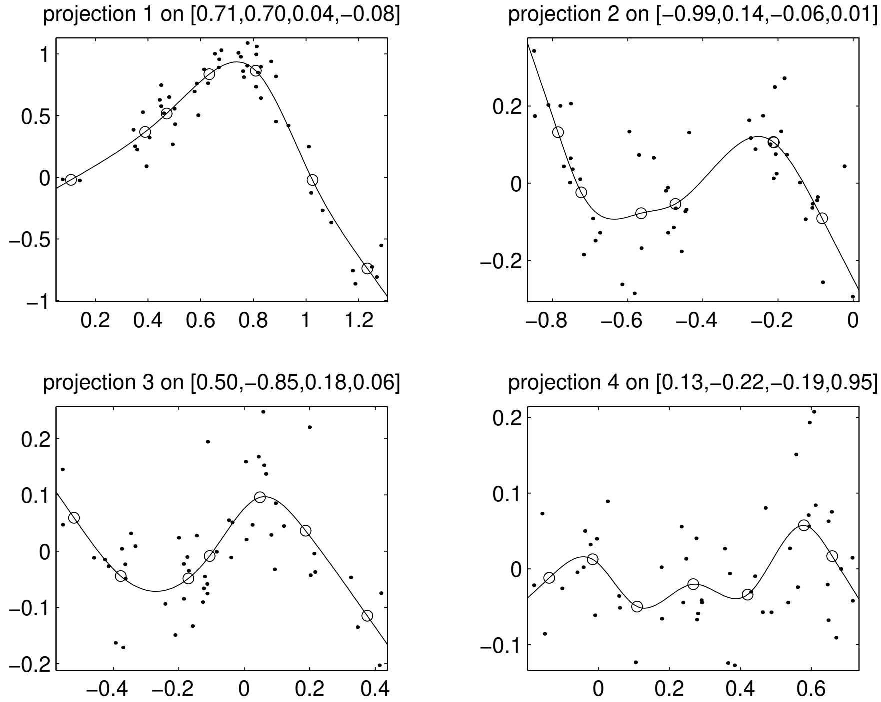

Figure 2: First 4 projections for $y = \sin(2\pi x_1 x_2) + \mathcal{N}(0, 0.1)$ obtained using projection pursuit with self-organizing maps for knot placement.

variables"), and Gaussian noise with standard deviation 0.1. We use a network with $n = 7$ knots, spline interpolation between these knots, and lateral interaction

$$h_{i+j,i} = h_{i-j,i} \propto \begin{cases} \exp\left[-\frac{j^2}{2\sigma^2}\right], & \text{if } i+j \leq n \text{ and } i-j \geq 1, \\ 0, & \text{if } i+j > n \text{ or } i-j < 1, \end{cases}$$

where for each i the interactions are normalized such that $\sum_j h_{ji} = 1$. Figure 2 shows the first 4 projections for $\sigma = 0.2$ and $\beta = 500$, corresponding to a solution of the form

$$\sin(2\pi x_1 x_2) \approx f_1(x_1 + x_2) + f_2(x_1) + f_3(x_2) + \ldots .$$

It can be seen that in this example overfitting starts with the fourth projection.

3 Conclusion

We have given a unified theoretical framework for the formation of topological maps. Specific choices for the noise parameters in these self-organizing maps yield algorithms for nonparametric regression. Future work should aim at improving these algorithms and comparing them with other suggestions. Possible interesting extensions are learning rules for smooth nonfunctional mappings, projection pursuit on more than one dimension, and mixture-like models for joint probability estimation with additional lateral interaction (see [5]).

References

[1] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[2] T. Heskes and B. Kappen. Error potentials for self-organization. In *International Conference on Neural Networks, San Francisco*, volume 3, pages 1219–1223, New York, 1993. IEEE.

[3] S. Luttrell. A Bayesian analysis of self-organizing maps. *Neural Computation*, 6:767–794, 1994.

[4] K. Rose, E. Gurewitz, and G. Fox. Statistical mechanics of phase transitions in clustering. *Physical Review Letters*, 65:945–948, 1990.

[5] B. Kappen. Deterministic learning rules for Boltzmann Machines. *Neural Networks, in press*, 1995.

[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[7] C. Peterson. Parallel distributed approaches to combinatorial optimization: benchmark studies on traveling salesman problem. *Neural Computation*, 2:261–269, 1990.

[8] V. Cherkassy and H. Lari-Najafi. Constrained topological mapping for nonparametric regression analysis. *Neural Networks*, 4:27–40, 1991.

[9] J. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Society*, 76:817–823, 1981.

[10] J. Hwang, S. Lay, M. Maechler, D. Martin, and J.. Schimert. Regression modeling in back-propagation and projection pursuit learning. *IEEE Transactions on Neural Networks*, 5:342–353, 1994.