ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 205-210

Linearization identification and an application to BSS using a SOM

F.J. Theis and E. Lang

Institute of Biophysics, University of Regensburg,
93040 Regensburg, Germany

Abstract. The one-dimensional functional equation $g(y(t)) = cg(z(t))$ with known functions y and z and constant c is considered. The indeterminacies are calculated, and an algorithm for approximating g given y and z at finitely many time instants is proposed. This *linearization identification* algorithm is applied to the postnonlinear blind source separation (BSS) problem in the case of independent sources with bounded densities. A self-organizing map (SOM) is used to approximate the boundary, and the postnonlinearity estimation in this multivariate case is reduced to the one-dimensional functional equation from above.

1 Introduction

Linearization identification solves the functional equation 1. Intuitively, it calculates g such that the two functions y and z are scalings of each other after application of g i.e. the linear relationship between $g \circ y$ and $g \circ z$ already determines g. It can be applied for separating postnonlinear mixtures given various restrictions to the sources. We demonstrate this in the case of independent bounded sources. Other possible applications include assumptions such as sparseness or nonnegativity of the sources.

The paper is organized as follows: In the next section, we calculate the indeterminacies of linearization identification and give an algorithm for estimating g. Section 3 applies this to postnonlinear BSS, and section 4 gives a computer simulation thereof.

2 Linearization identification

Linearization identification describes the problem of finding a diffeomorphism g on subsets of $\mathbb{R}$ with

$$g \circ y = cg \circ z \tag{1}$$

for an unknown constant $c \neq \pm 1$ and given continuously differentiable functions $y, z : (-\varepsilon, \varepsilon) \to \mathbb{R}$ with $y(0) = z(0) = 0$ and $y'(0), z'(0) \neq 0$.

2.1 Indeterminacies

First note that c is determined by y and z. Indeed by taking derivatives in equation 1, we get $c = y'(0)/z'(0)$, hence $c \neq 0$. Furthermore $g(0) = 0$ because otherwise $c = 1$.

If g is a solution of the linearization identification problem, then so is λg for $\lambda \neq 0$. So scaling is an obvious indeterminacy of the above. The following theorem shows that there are no more indeterminacies.

Theorem 2.1. *If equation 1 has an analytic solution, then it is unique (among all analytical solutions) except for scaling.*

We assume analyticity only for the sake of simplicity; in fact this result can be generalized for only continuously differentiable functions.

Proof. Let g and $\bar{g}$ solve equation 1. Then $y = \bar{g}^{-1} \circ (c\bar{g}) \circ z$, hence $g \circ \bar{g}^{-1}(c\bar{g}) \circ z = cg \circ z$. Setting $h := \bar{g} \circ g^{-1}$ and $\bar{z} = g \circ z$, this guarantees $h \circ (ch) \circ \bar{z} = c\bar{z}$ or $ch(\bar{z}) = h(c\bar{z})$. Taking derivatives and dividing by $c\bar{z}$ (nonzero locally at 0) yields $h'(\bar{z}) = h'(c\bar{z})$ locally at 0 and then inductively $h^{(n)}(\bar{z}) = c^{n-1}h^{(n)}(\bar{z})$, where $h^{(n)}$ denotes the n-th derivative of h. So $h^{(n)}(0) = 0$ for $n > 1$ because $c \neq \pm 1$. But by assumption h is analytic, so h is already linear, say $h(t) = \lambda t$. Then $\bar{g} = \lambda g$ as was to show. □

Later we will have to solve equation 1 for functions y and z defined only on $[0, \varepsilon)$; then we assume that the right differentials at 0 exist and fulfill the above. Uniqueness in this case follows from the above theorem.

2.2 Algorithm

The goal of this section is to find an interpolation approach in order to solve the linearization identification problem numerically, given samples $y(t_1), z(t_1), \ldots, y(t_T), z(t_T)$ of the 'curves' y and z. In the following, we present an algorithm based on polynomial approximation using least-squares estimation. Other estimation algorithms are possible — we get successful results with derivative based polynomial approximation and multilayer-perceptron approximation of g with natural gradient descent on $E(g)$ from below — but are omitted here due to limited space.

Note that we here assume that the samples of y and z are given at the *same* time instants $t_i \in (-1, 1)$. In practice, this is usually not the case, so values of z at the sample points of y and vice versa will first have to be estimated, for example by using spline interpolation.

Furthermore in order to determine the constant c estimates of the derivative of y and z at 0 are needed. Without loss of generality let $t_1 < \ldots < t_T$ and $t_{i-1} < 0 < t_i$. Then using $y(0) = 0$

$$c = y'(0)/z'(0) \approx \frac{y(t_i) - y(t_{i-1})}{z(t_i) - z(t_{i-1})} \approx \frac{y(t_i)}{z(t_i)}$$

depending on whether single-side or both-side secant derivative estimation is to be used. Therefore, in equation 1 only g is to be estimated.

We want to find an approximation $\tilde{g}$ (in some parametrization) of g with with $\tilde{g}(y(t_i)) = c\tilde{g}(z(t_i))$ for $i = 1, \ldots, T$, so in the most general sense we want to find

$$\tilde{g} = \operatorname{argmin}_g \sum_{i=1}^{T} (g(y(t_i)) - cg(z(t_i)))^2 =: \operatorname{argmin}_g E(g). \tag{2}$$

As in theorem 2.1 we will assume that g is analytic and show how to approximate g in this case. In order to estimate g, we can expand it locally at 0 into a convergent power series $g(t) = \sum_{i=0}^{\infty} a_i t^i$, with $a_i = \frac{1}{i!} g^{(i)}(0)$. g can now be approximated as closely as desired (with error bounds given by Taylor's formula) by only taking the Taylor expansion to a certain degree d: $\tilde{g}(t) := \sum_{i=0}^{d} \tilde{a}_i t^i$. We know $\tilde{g}(0) = 0$, and in order to eliminate the permutation indeterminacy, we can furthermore set $\tilde{g}'(0) = 1$, so the $d - 1$ parameters $\tilde{a}_2, \ldots, \tilde{a}_d$ are to be estimated from equation 1.

Putting this polynomial approximation into equation 2, the 'energy' function to be minimized can be calculated as

$$E(g) = \sum_{i=1}^{T} \left(\sum_{j=2}^{d} (y(t_i)^j - cz(t_i)^j) a_j - (cz(t_i) - y(t_i)) \right)^2$$

This can obviously be transformed into the usual least-squares form by defining a $T \times (d-1)-$matrix $\mathbf{M}$ with entries $M_{ij} := y(t_i)^j - cz(t_i)^j$ and a T-dimensional vector $\mathbf{v}$ with $v_i := cz(t_i) - y(t_i)$. Then the energy function can be written as $E(g) = \|\mathbf{Ma} - \mathbf{v}\|^2$ with the coefficient vector $\mathbf{a} = (a_2, \ldots, a_d)^\top$ where $\|.\|$ denotes the Euclidean norm. Equation 2 implies finding the polynomial $\tilde{g}$ with coefficients $\tilde{\mathbf{a}}$ such that

$$\tilde{\mathbf{a}} = \operatorname{argmin}_{\mathbf{a}} E(g) = \operatorname{argmin}_{\mathbf{a}} \|\mathbf{Ma} - \mathbf{v}\|^2.$$

We can assume that $T \geq d - 1$. Then the above equation is well-known to have the unique solution $\tilde{\mathbf{a}} = \mathbf{M}^+ \mathbf{v}$ where $\mathbf{M}^+$ denotes the pseudo inverse of $\mathbf{M}$. Note that instead of polynomial approximation also other *linearly* parametrized models can be used for $\tilde{g}$ in exactly the same manner.

3 Postnonlinear ICA of bounded sources

The aim of independent component analysis (ICA) is to find statistically independent data within a given random vector. An application of ICA lies in blind source separation (BSS), where it is furthermore assumed that the given vector has been mixed using a fixed set of independent sources. Good textbook-level introductions to ICA are given in [2,3].

We call a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ *diagonal* if each component $f_i(\mathbf{x})$ of $\mathbf{f}(\mathbf{x})$ depends only on the variable x_i. In this case we often omit the other variables and write $\mathbf{f}(x_1, \ldots, x_n) = \big(f_1(x_1), \ldots, f_n(x_n)\big)$.

Consider the postnonlinear BSS model $\mathbf{X} = \mathbf{f}(\mathbf{AS})$, where $\mathbf{S}$ is an n-dimensional independent random vector, $\mathbf{A}$ an invertible real matrix and $\mathbf{f}$ a diagonal nonlinearity. We assume that $\mathbf{S}$ (and hence $\mathbf{X}$) has a bounded density and that the components f_i of $\mathbf{f}$ are injective analytic functions with non-vanishing derivative. Then also f_i^{-1} is analytic. It is known that this model is separable given some further restrictions to the mixing matrix $\mathbf{A}$, which are fulfilled by almost all matrices in the measure sense [6].

An algorithm for finding the model and the sources given only $\mathbf{X}$ is presented in [1] for two dimensions, based on spline interpolation of the boundary and Hudson minimization of a joint energy function. In the following we will show that the underlying postnonlinearity detection in arbitrary dimensions can be reduced to the one-dimensional linearization identification problem from above; this enables us to separate postnonlinearily mixed data using the polynomial least-squares algorithm from section 2.2.

The separation is done in a three-stage procedure: In the first step, we preprocess the data in the sense that we detect submanifolds in the mixture data that are mapped onto lines by the nonlinearity. The second step estimates the separating postnonlinearities using linearization identification. In the final stage, the mixing matrix $\mathbf{A}$ and then the sources $\mathbf{S}$ are reconstructed by applying linear ICA to the linearized mixtures $\mathbf{f}^{-1}\mathbf{X}$.

3.1 Boundary detection

Let $\mathbf{x}(1), \ldots, \mathbf{x}(T') \in \mathbb{R}^n$ be i.i.d.-samples of the random vector $\mathbf{X}$. We want to construct vectors $\mathbf{y}(1), \ldots, \mathbf{y}(T)$ and $\mathbf{z}(1), \ldots, \mathbf{y}(T) \in \mathbb{R}^n$ using clustering or interpolation on the samples $\mathbf{x}(t)$ such that $\mathbf{f}^{-1}(\mathbf{y}(t))$ and $\mathbf{f}^{-1}(\mathbf{z}(t))$ lie in two linearly independent one-dimensional subvectorspaces of $\mathbb{R}^m$.

At first the mixture data $\mathbf{X}$ is preprocessed in order to highlight the boundary $\partial\mathbf{X}$. If the density $p_{\mathbf{X}}$ of $\mathbf{X}$ is assumed to have greatest ascent at the boundary, this can be done by estimating the gradient $\nabla p_{\mathbf{X}}$ and thresholding its norm. For estimating the density $p_{\mathbf{X}}$ given samples $\mathbf{x}(1), \ldots, \mathbf{x}(T)$ we use kernel density estimation $\hat{p}_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Tr^n}\sum_{t=1}^{T} K\left(\frac{\mathbf{x}-\mathbf{x}(t)}{r}\right)$ with kernel radius r and the Epanechnikov kernel

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2c_n}(n+2)(1-\|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise} \end{cases}$$

where c_n denotes the volume of the n-dimensional unit sphere [5]. This kernel is optimal in the sense that it yields minimum mean integrated square error. The derivative of the density can then be estimated as

$$\nabla \hat{p}_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Tr^n}\sum_{t=1}^{T} \nabla K\left(\frac{\mathbf{x}-\mathbf{x}(t)}{r}\right) = \frac{n+2}{Tr^{n+2}c_n}\sum_{\|\mathbf{x}(t)-\mathbf{x}\|<r} \mathbf{x}-\mathbf{x}(t)$$

Taking only samples $\mathbf{x}(t)$ with $\|\nabla\hat{p}_{\mathbf{X}}(\mathbf{x}(t))\| > \theta$ for an appropriate threshold $\theta > 0$, we can estimate the mixture density boundary under the above assumption.

In the next step, the boundary submanifolds are estimated using a *selforganizing map (SOM)* algorithm, a clustering algorithm often used for the visualization of high-dimensional data. SOMs have been developed by Kohonen in 1981 [4] and have since then become a widely used and studied visualization and clustering technique.

We simply approximate $\partial\mathbf{X}$ using a SOM with neuron structure $\{1, \ldots, k\}^n$ (rectangular topology in two dimensions). After training, neurons with 1 or k at some index position give approximations of the boundary. In practice it is advisable to prewhiten the data before applying the SOM algorithm [7].

Other algorithms for estimating boundary lines are possible — for example similar to histogram based density estimation, the boundaries can be approximated in a piecewise linear fashion by discretizing the space in each coordinate direction and taking minima and maxima; this is done in the two-dimensional case in [1]. The advantage of the presented method is that we use standard algorithms and do not have to estimate the boundary corners separately — this is done by the rectangular structure of the SOM automatically.

3.2 Postnonlinearity estimation

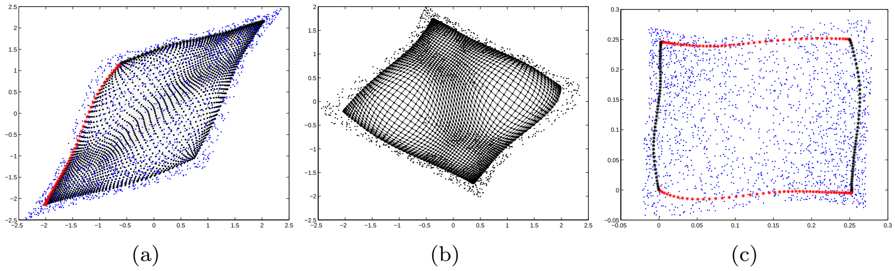
Given the subspace vectors $\mathbf{y}(t)$ and $\mathbf{z}(t)$ from the previous section, the goal is to find diffeomorphisms $g_i : \mathbb{R} \to \mathbb{R}$ such that $g_1 \times \ldots \times g_m$ maps the vectors $\mathbf{y}(t)$ and $\mathbf{z}(t)$ onto two different *linear* subspaces.

In abuse of notation, we now assume that two *functions* $\mathbf{y}, \mathbf{z} : [0, 1] \to \mathbb{R}^m$ are given with $\mathbf{y}(0) = \mathbf{z}(0) = 0$. These can for example be constructed from the discrete sample points $\mathbf{y}(t)$ and $\mathbf{z}(t)$ from the previous section by polynomial or spline interpolation.

Let $i \neq j$ be fixed. In [6] it was shown that the analytical nonlinearities $f_i \times f_j$ are already uniquely determined (except for scaling) by knowing the images of two nonlinearly transformed lines under the postnonlinear mapping $f_i \times f_j(\pi_{ij}\mathbf{A})$, where π_{ij} denotes the projection onto the i-th and j-th coordinate. So in fact after projection of $\mathbf{y}(t)$ and $\mathbf{z}(t)$ using π_{ij} it is sufficient to consider the case $m = 2$ and g_2 is to be reconstructed, which we will assume in the following.

We can assume that the indices i, j were chosen such that the two lines $\mathbf{f}^{-1} \circ \mathbf{y}, \mathbf{f}^{-1} \circ \mathbf{z} : [0, 1] \to \mathbb{R}^2$ do not coincide with the coordinate axes. Reparametrization ($\bar{\mathbf{y}} := \mathbf{y} \circ \mathbf{y}_1^{-1}$) of the curves lets us further assume that $\mathbf{y}_1 = \mathbf{z}_1 = \mathrm{id}$. Then the condition that the separating nonlinearities $\mathbf{g} = g_1 \times g_2$ must map $\mathbf{y}$ and $\mathbf{z}$ onto lines can be written as $\mathbf{g} \circ \mathbf{y} = (g_1, ag_1)$ and $\mathbf{g} \circ \mathbf{z} = (g_1, bg_1)$ with constants $a, b \in \mathbb{R} \setminus \{0\}$, $a \neq \pm b$. This is equivalent to $g_2 \circ y_2 = ag_1 = \frac{a}{b} g_2 \circ z_2$.

So determining g_2 (and similarly g_1) is equivalent to performing linearization identification for g_2 using the functions y_2 and z_2. Theorem 2.1 shows that in this case g_2 is uniquely determined by y_2 and z_2 (this gives an indication how to prove separability in the analytical case [6]) and the least-squares algorithm from section 2.2 can be used to estimate g_2 (hence $\mathbf{g}$) from sample data.



(a)

(b)

(c)

Figure 1: Example 2. A postnonlinear mixture of two uniform signals is separated. In (a), the mixture density of $\mathbf{X}$ together with the trained SOM for the boundary estimation is shown. (b) gives the whitened $\partial\mathbf{X}$ with the SOM, and (c) depicts the recovered source density together with the SOM boundaries.

4 Simulation

In order to demonstrate the algorithm, in $[-1,1]^2$ uniform data (2000 samples) is mixed first linearly and then nonlinearly using the postnonlinearity $\mathbf{f}(\mathbf{x}) = (\tanh(x_1)+x_1, \tanh(2x_2)+x_2)$. A 50×50 rectangular SOM was used to approach the mixture boundary (approximated using $r = 1, \theta = 0.05$) in 10 epochs. The learnt polynomials of degree 10 were rather linear, and we were able to recover the sources with SNRs of 22 and 13 dB with respect to the original sources, see fig. 1. This error could be minimized by approximating the boundary with other algorithms: if we use the real (calculated) boundary, we get recovery SNRs of 30 and 22 dB. Still, also in this case, the polynomials were not fully able to approximate the nonlinearities. The performance can be enhanced by not only using two lines of the mixtures but the whole boundary, which leads to a similar least-squares optimization as above.

References

[1] M. Babaie-Zadeh, C. Jutten, and K. Nayebi. A geometric approach for separating post non-linear mixtures. In *Proc. of EUSIPCO '02*, volume II, pages 11–14, Toulouse, France, September 2002.

[2] A. Cichocki and S. Amari. *Adaptive blind signal and image processing.* John Wiley & Sons, 2002.

[3] A. Hyvärinen, J.Karhunen, and E.Oja. *Independent Component Analysis.* John Wiley & Sons, 2001.

[4] Teuvo Kohonen. Self-organizing formation of topologically correct feature maps. *Biol. Cyb.*, 43(1):59–69, 1982.

[5] B.W. Silverman. *Density Estimation for Statistics and Data Analysis.* New York: Chapman and Hall, 1986.

[6] F.J. Theis and P. Gruber. Separability of analytic postnonlinear blind source separation with bounded sources. ESANN 2004, 2004.

[7] F.J. Theis, C. Puntonet, and E.W. Lang. SOMICA - an application of self-organizing maps to geometric independent component analysis. In *Proc. of IJCNN 2003*, pages 1318–1323, Portland, USA, 2003.